APPLIED COMPUTER SCIENCE

The Journal is a peer-reviewed, international, multidisciplinary journal covering a broad spectrum of topics of computer application in production engineering, technology, management and economy.

The main purpose of Applied Computer Science is to publish the results of cutting-edge research advancing the concepts, theories and implementation of novel solutions in computer technology. Papers presenting original research results related to applications of computer technology in production engineering, management, economy and technology are welcomed.

We welcome original papers written in English. The Journal also publishes technical briefs, discussions of previously published papers, book reviews, and editorials. Especially we welcome papers which deals with the problem of computer applications in such areas as:

- manufacturing,
- engineering,
- technology,
- designing,
- organization,
- management,
- economics,
- innovations,
- competitiveness,
- quality and costs.

The Journal is published quarterly and is indexed in: BazTech, Cabell's Directory, Central & Eastern European Academic Source (CEEAS), CNKI Scholar (China National Knowledge Infrastucture), DOAJ (Directory of Open Access Journals), EBSCO, ERIH PLUS, Index Copernicus, J-Gate, Google Scholar, Scope Database, Scopus, TEMA Technik und Management.

Letters to the Editor-in-Chief or Editorial Secretary are highly encouraged.

CONTENTS

Applied Computer Science, vol. 17, no. 4, pp. 5–19
doi: 10.23743/acs-2021-25

Submitted: 2021-10-01 | Revised: 2021-11-09 | Accepted: 2021-12-08

Keywords: electromobility, powertrain, electric drives, artificial neural network, efficiency modelling

Lukas BAUER [0000-0002-1641-0376]*, *Leon STÜTZ* [0000-0002-4754-3251]* , *Markus KLEY* [0000-0003-4061-0797]*

BLACK BOX EFFICIENCY MODELLING OF AN ELECTRIC DRIVE UNIT UTILIZING METHODS OF MACHINE LEARNING

Abstract

The increasing electrification of powertrains leads to increased demands for the test technology to ensure the required functions. For conventional test rigs in particular, it is necessary to have knowledge of the test technology's capabilities that can be applied in practical testing. Modelling enables early knowledge of the test rigs dynamic capabilities and the feasibility of planned testing scenarios. This paper describes the modelling of complex subsystems by experimental modelling with artificial neural networks taking transmission efficiency as an example. For data generation, the experimental design and execution is described. The generated data is pre-processed with suitable methods and optimized for the neural networks. Modelling is executed with different variants of the inputs as well as different algorithms. The variants compare and compete with each other. The most suitable variant is validated using statistical methods and other adequate techniques. The result represents reality well and enables the performance investigation of the test systems in a realistic manner.

1. INTRODUCTION

The steadily advancing climate change requires a reduction of greenhouse gases. With annual emissions of 160,000 kilotons of CO_2 equivalent, the transportation sector in Germany has great potential for savings (German Environment Agency, 2020). As a result, there is a demand for and promotion of climate-friendly solutions for mobility. The electrification of vehicles is a key tool for the reduction of CO_2 emissions (Hoekstra, 2019). Currently, a corresponding increase in electric drives is discernible. Still, the potential for climate-neutral transportation also brings challenges in vehicle technology. The functionality of electric powertrains must be validated on test rigs. Increased dynamics of electric powertrains must be considered. Knowledge of the physical limits of the test rig, like the dynamics, is not always available. Especially with conventional test rigs, the required dynamics may exceed the accessible ones. In the best case, the planned test setup can then only be carried out by means of an adaptation, in the worst case, not at all.

* Aalen University, Institute for Drive Technology, Germany, lukas.bauer@hs-aalen.de, leon.stuetz@hs-aalen.de, markus.kley@hs-aalen.de

The use of testing technology models enables an early statement about the feasibility of the tests. In doing so, the test rig can be modelled in its entirety (Bauer, Bauer & Kley, 2021). The model includes the electrics, the mechanics as well as the control system. By means of white-box modelling, the model is represented theoretically and idealized. Complex processes, such as friction, can hardly be represented by theoretical modelling. They can be represented using the conventional approaches of experimental modelling, such as the description by characteristic maps and curves (Stütz, Beck & Kley, 2021). The calculation is performed using measured data at specified operating points. Between the measured operating points, the data is interpolated. Patterns in the measured data are recognized only sparsely. Due to the necessary intermediate steps for the calculation, modelling with maps also has a relatively low computational efficiency.

By using innovative approaches for modelling, the available information content of the measured data as well as the computational efficiency can be increased (Bauer, Beck, Stütz & Kley, 2021). The aim of this work is to provide an experimental model of the efficiency of an electrical machine. The result is a function that continuously describes the efficiency based on the mechanical inputs speed and torque. For this purpose, the mechanical, electrical and thermal influences on the system are investigated in detail. The procedure for modelling the function is described in detail. Suitable experiments are planned and carried out to generate the data. The measured data is pre-processed by suitable methods. The model is built using artificial neural networks (ANN). To optimize the model, different inputs as well as different training algorithms are compared. The result is validated by different approaches, such as a statistical evaluation of the error.

2. STATE OF THE ART

The product creation process is described in particular with the help of the V-Model (see Fig. 1). Simulations can already be carried out in the component specification. These enable tests without a physical Device Under Test (DUT). (Stütz, Bauer & Kley, 2019)

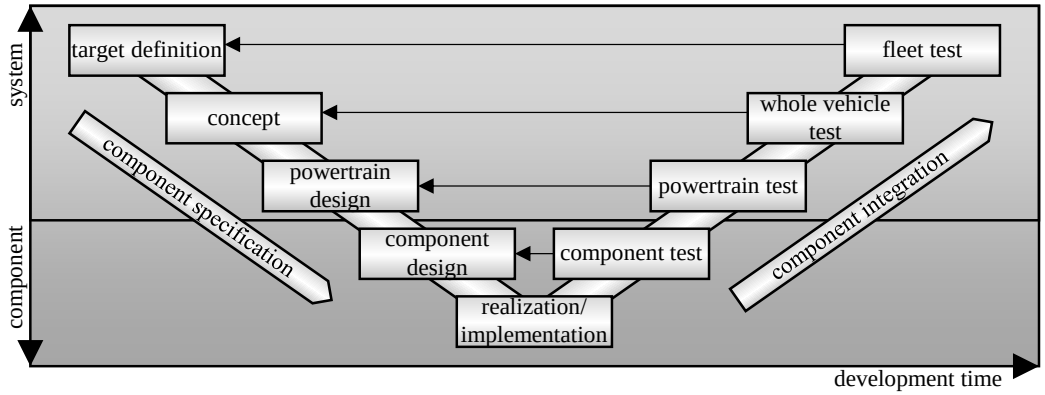


Fig. 1. V-Model based on (Dohmen, Pfeiffer & Schyr, 2009) and (Paulweber & Lebert, 2014)

As time increases during the product development process, so does the number of available physical DUTs. This enables increasing, real testing in the component integration. As the number of available DUTs increases, the scope of the tested systems increases, for example from individual components to the complete powertrain. Finally, the complete vehicle is tested on the road. Any necessary changes are implemented iteratively in optimisation loops. (Dohmen, Pfeiffer & Schyr, 2009)

A minimum value of kilometres to be driven is required to fully validate product features. Typical values for the example of autonomous driving range from one million kilometres for the validation of a function to one billion kilometres for the complete validation of the system. A corresponding safeguarding by driving on the road is hardly feasible in the available development time. A procedure with increasing relevance is therefore testing on the test rig. (Beine & Rasche, 2018).

Early validation on the test rig or through simulation also keeps product development costs low (Albers, Behrendt, Klingler & Matros, 2016).

2.1. Efficiency Mapping

Test rig experiments for efficiency testing provide knowledge about the usable energy share of a system. During power difference measurement, the input power P_{in} and the output power P_{out} are determined. From this, the efficiency η is derived according to formula (1):

$$\eta = \frac{P_{out}}{P_{in}} \tag{1}$$

where: η – efficiency,
P_{out} – output power,
P_{in} – input power.

The power of the drive unit can be determined, for example, by torque M_{out} and speed n_{out} sensors on the output shaft or current I and voltage U sensors in the armature of the electric machine. With the power difference measurement, the efficiency in the motor operation can be calculated according to formula (2):

$$\eta = \frac{P_{out}}{P_{in}} = \frac{2\pi \cdot M_{out} \cdot n_{out}}{U * I} \tag{2}$$

where: M_{out} – output torque,
n_{out} – output speed,
U – armature voltage,
I – armature current.

2.2. Modelling

Currently, there is also an increasing trend to shift testing assignments from road and test rig applications to simulation software. Nevertheless, it can be assumed that physical testing will be necessary for the foreseeable future to ensure the required product properties (Dismon, 2017). It is not to be seen as a competitor to simulation, but as an extension to be

used synergetically in order to achieve optimized results (Guggenmos, Rückert, Thalmair & Wagner, 2015). Willmerding & Häckh (2017) for example, describe the combination of vehicle simulation and test rig control for mapping highly dynamic driving cycles on test rigs. The winEVA tool used thus enables more realistic results through driving maneuver-based test scenarios. The validation on the test rig can be optimized by using such numerical tools.

The simulation of scenarios on the computer requires models that describe the planned or real system. Isermann (2008) principally distinguishes between the theoretical and the experimental modelling. The main kinds of modelling as well the intermediate stage grey-box model are displayed in Fig. 2.

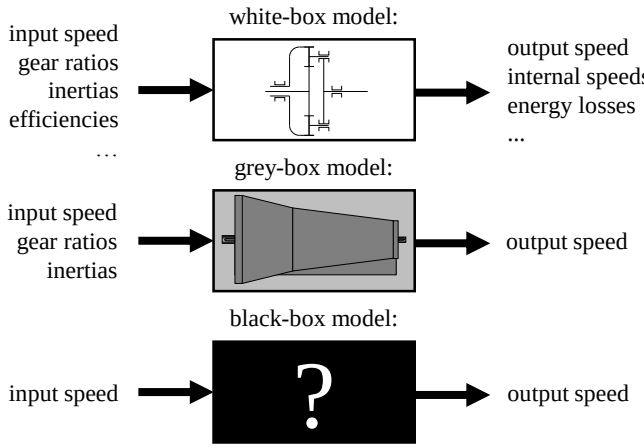


Fig. 2. Modelling concepts

Theoretical modelling of white-box models requires comprehensive knowledge of the system to be modelled. The overall system is usually subdivided into smaller subsystems. The processes and relationships are described physically. The theoretical modelling for the efficiency of a transmission using a simulation is described, for example, by Li et al. (2014) and Ratov & Lyfar (2020).

In experimental modelling of black-box models, also referred to as parameter identification, the processes in the system are not known. It is described exclusively with the input and output variables as well as its transfer function. There are various ways to establish the relationship between the variables like the classic methods of using characteristic diagrams or the calculation of polynomials. In addition to the classical methods, modelling with artificial neural networks is increasingly used. Machrowska et al. (2020) compare the mathematical modelling with polynomials to the modelling with ANNs. The ANNs lead to superior results. The creation of a transfer function for recognition and depiction of correlations between input and output signals is one of the basic ideas of ANNs. Therefore, they're suitable for the modelling of complex systems. There is a wide range of possible applications in modelling. Khan et al. (2020) describe the creation of an efficiency function of an electrical machine by means of ANNs. The training data originates from numerical calculations. Çelik et al. (2017) describe the creation of an efficiency and power function of an electric machine. The training data results from measurements.

Yadav & Yadava (2017) describe the modelling of an ANN for an EAHM process. A parameter study was carried out for optimization. sufficiently good results were achieved with the scaled-conjugate-gradient algorithm. Payal et al. (2013) compare and compete the Levenberg-Marquardt algorithm and the Bayesian-Regularization algorithm. The algorithms are applied for the efficient localization in wireless sensor networks. It is concluded, that the Bayesian-Regularization algorithm produces more accurate results at the expense of higher training time, than the Levenberg-Marquardt algorithm. Based on this Jazayeri et al. (2016) compare and compete the Levenberg-Marquardt algorithm and the Bayesian-Regularization algorithm for power estimation of photovoltaic modules. The better accuracy at the expense of higher training time by using the Bayesian-Regularization algorithm can also be approved for the described use case. In conclusion both papers recommend to use the Levenberg-Marquardt algorithm for time critical applications. Otherwise the Bayesian-Regularization algorithm is recommended. Based on this, this paper investigates the suitability of the described algorithms for modelling the efficiency.

3. PARAMETER IDENTIFICATION

The investigation is carried out using a drive test rig with a power rating of 300 kW. The test rig consists of a prime mover, which simulates the vehicle drive controlled by its speed and a load machine, which simulates the driving resistances controlled by its torque. The electrical machines are externally excited DC motors. In order to achieve a sufficiently high speed, transmissions are connected to the motors. The input uses a non-switchable planetary gear with a ratio of 3.2. The output uses a switchable planetary gear with ratios $i_1 = 1$ and $i_2 = 3.47$. Drive train components such as transmissions can be connected and tested between the two motors.

As a basis for the project, a digital twin of the described test rig was developed by Bauer et al. (2021). This primarily represents the dynamic behavior of the given test rig. This allows planned test scenarios to be examined in advance for their feasibility and optimized for the given test technology. Up to now, the modelling has been done as a white-box model. This is an ideal, loss-free representation of the system. Factors such as power loss are not considered. For a more realistic simulation of the real system, the aim is to model the complex efficiency. This is to be done by functions from ANNs. Training data that can be processed as the basis of the ANNs are necessary for model building. For this purpose, an experimental design is carried out and measurement data is recorded through experiments. The resulting raw data is preprocessed for training the ANN. With the help of existing algorithms, the networks are trained on the given data and can be integrated into simulations. The results are validated in particular by comparing them with the real system. The procedure is described in detail below.

3.1. Experimental design and data generation

To generate data, appropriate tests are run on the real test rig. Due to its inferior performance, the investigation is limited to the drive for the time being. For data generation, the two machines are connected directly via a constant velocity drive shaft and operated without a DUT. This avoids disturbing influences of the DUT.

For data acquisition, a large number of measured values are recorded at different measuring points. These are differentiated into mechanical, electrical and thermal.

The distribution of the measuring points T_1, T_2, T_3 and T_4 for the temperature is based on infrared images of the machine in operation. This allows points to be identified which react particularly quickly to the heating of the machine. The test setup with the measuring points and the comparison with the infrared images is shown in Fig. 3.

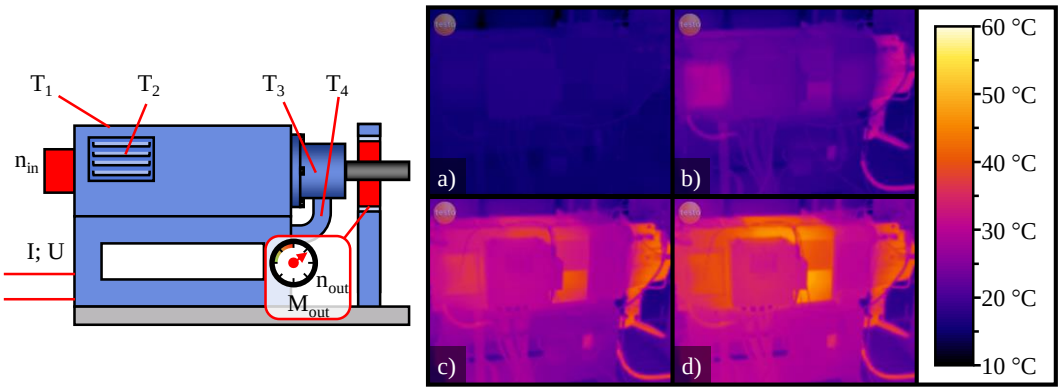


Fig. 3. Schematic illustration of the measurement locations and IR-illustration of the heating after: a) 0 min, b) 50 min, c) 100 min and d) 150 min

The aim of the experiments is to generate data for which an ANN can recognize correlations with the efficiency. This is to be described by a function from a trained network. The collective used for this consists of quasi-randomly distributed points in the machine's characteristic diagram.

This is to ensure that no patterns are given to the network and that it recognizes any given patterns itself. In addition, the measuring points are approached in random order to take the temperature influence into account. As a result, the temperature warms up independently of the operating point. The principle is shown in Fig. 4.

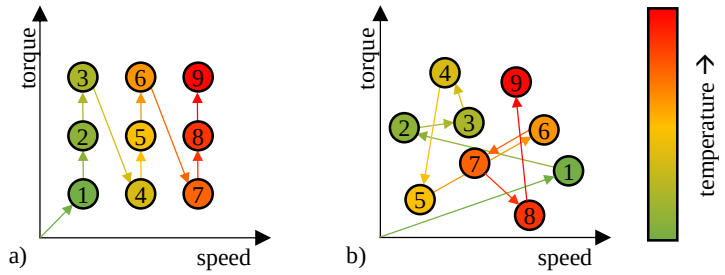


Fig. 4. Measurement point distribution with: a) uniformly sequential and b) random points

The start-up time of the individual operating points is determined as a function of the speed difference. A maximum speed ramp is specified. A time buffer of 5 seconds is also added. After start-up, the points are held for 30 seconds. This results in an essential division of the measurement data into dynamic start-up and static holding of the points.

The collective is approached for about five hours. At a measurement frequency of 10 Hz, about 190,000 data samples are recorded. Through a correlation analysis between the efficiency and each individual temperature channel, no significant influence of the temperature on the overall efficiency can be determined. It can be assumed that as the temperature increases, the transmission efficiency increases, but the motor efficiency decreases. It is assumed that a superposition of the individual efficiency changes keeps the overall influence of the temperature low.

3.2. Data pre-processing

For data pre-processing, relevant measured values are selected in advance for the input. These are in particular mechanical values such as the speed and torque. In addition to the main values, the influence of derived values is also examined. These are in particular the torque gradient and the speed gradient.

The efficiency of the machine is used for the ANN output. This is determined on the basis of the measured values. The measurement data is limited to motor forward operation. It can be assumed that the results can be mirrored to other operating modes.

Some of the data contain information that apparently has no plausible information value. For example, in the area of the dynamic start-up of the measuring points, there is a strong fluctuation of the current around the expected value with a constant period of T = 0.5 s (see Fig. 5).

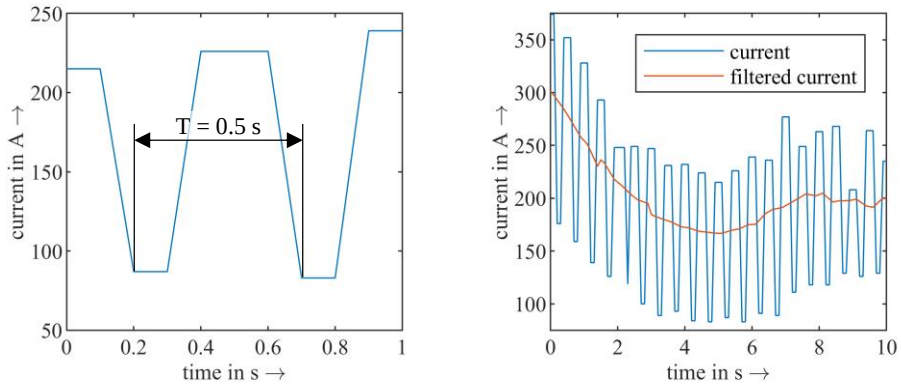


Fig. 5. Fluctuations in the current and raw and filtered current signal

The fluctuations are transferred to all values derived from the current, but not to the mechanics. Detailed investigations into the cause of the discrepancy, such as an analysis of the frequency response of the mechanical oscillations, did not yield any usable outcome. To avoid a resulting degradation of the results, the signal is smoothed with a filter. According to a correlation analysis a moving average with a window of $3*T$ = 1.5 s provides the best results. The consequential values in operation without field weakening run approximately proportional to the measured torque.

The smoothing in the current is transferred to all values derived from it. The resulting signals, for example the target signal efficiency, are evaluated as plausible.

To avoid an unequal effect of the inputs, the selected data for input and output are normalized by their maximum values. Thus, all values lie in the numerical range [0,1]. The normalization is performed according to formula 4–7:

$$n_n = \frac{n}{\max(n)} \tag{4}$$

$$M_n = \frac{M}{\max(M)} \tag{5}$$

$$U_n = \frac{U}{\max(U)} \tag{6}$$

$$I_n = \frac{I}{\max(I)} \tag{7}$$

where: n_n – normalized rotational speed,
M_n – normalized torque,
U_n – normalized voltage,
I_n – normalized current.

The output signal efficiency cannot exceed the range [0,1] by definition. A normalization is therefore not needed.

3.3. Training of the artificial neural network

The network is a fully-connected multilayer perceptron. It consists of an input layer, two hidden layers and an output layer. The input layer has four neurons. The first hidden layer has 6 neurons, the second hidden layer has 4 neurons. The number of neurons in the hidden layers was determined empirically. The output layer has one neuron. It provides the output in the form of the efficiency. The network is shown in Fig. 6.

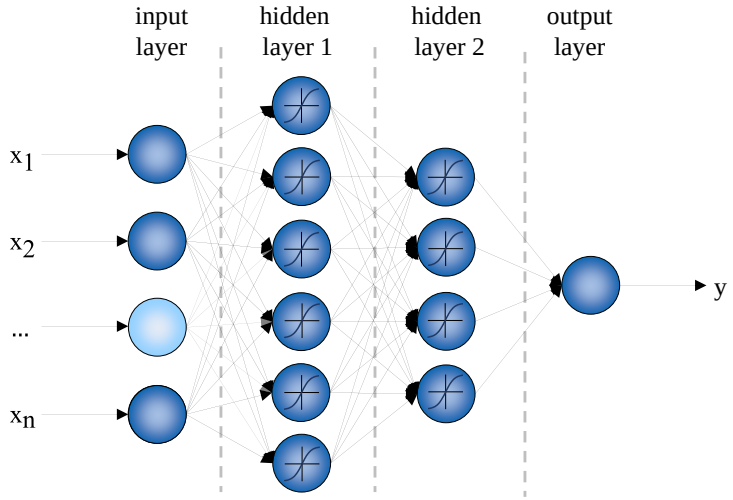


Fig. 6. Network architecture

The calculations, training and testing are done using Matlab. The training data is automatically and randomly divided into training data (70%), validation data (15%) and testing data (15%). The validation data is used to measure network generalization and halt the training when generalization stops improving while the testing data is used to measure the network performance during and after the training (The MathWorks, 2020).

Different versions of the network are created. The distinction is made in two categories. For the first category, the inputs are varied to determine the most suitable combination. In the second category, the training algorithms are varied. This is to determine the most suitable algorithm for the application.

The input variations are different combinations of speed, torque as well as their corresponding gradients. The variants have the following inputs:

- Variant 1: speed, torque.
- Variant 2: speed, torque, speed gradient.
- Variant 3: speed, torque, torque gradient.
- Variant 4: speed, torque, speed gradient, torque gradient.

The default set and frequently recommended Levenberg-Marquardt algorithm is used to determine the most suitable input variant (Jazayeri, Jazayeri & Uysal, 2016; Payal, Rai & Reddy, 2013). The training is partly random and therefore does not deliver uniform results. For a significant outcome, the training is performed 20 times per input variant. The characteristic values are presented in Fig. 7.

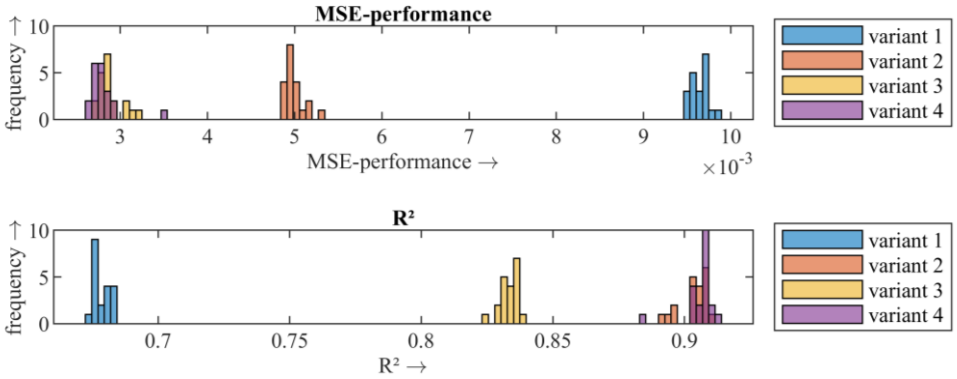


Fig. 7. Performance metrics with varying inputs

Analogous to Jazayeri et al. (2016), the resulting functions are evaluated on the basis of the performance metrics achieved. A relatively small deviation can be seen in the data. In terms of MSE performance, variant 3 and variant 4 achieve the lowest and thus the best values. For the R^2 value, variant 4 and variant 2 achieve the highest and thus the best values. Thus, variant 4 shows very good values in the essential criteria for the quality of results.

A summary of further averaged parameters is shown in Tab. 1. The values highlighted in green are the best. The values highlighted in red are the worst. In addition to the quality of the results, a good calculation time is recognizable for variant 4. Therefore, variant 4 is chosen as standard input for the further process.

Tab. 1. Performance metrics of varying inputs

Performance metrics	Variant 1	Variant 2	Variant 3	Variant 4
Best training performance (preferably low)	0.0096	0.0050	0.0029	0.0028
Best validation performance (preferable low)	0.0097	0.0050	0.0029	0.0028
Best testing performance (preferably low)	0.0097	0.0051	0.0029	0.0029
No. Of training epochs (preferably low)	173	196	221	175
R^2-value (preferably high)	0.678	0.904	0.833	0.906
Minimum gradient (preferably low)	5.16E-05	5.34E-05	1.53E-04	1.42E-04
Training time in s (preferably low)	32	39	43	36

The determination of a fitting algorithm for the ANNs is done by varying different approaches. The following algorithms are used for that:

- Levenberg-Marquardt algorithm (LM),
- Bayesian regularization algorithm (BR),
- Scaled-conjugate-gradient algorithm (SCG).

Each algorithm is trained 20 times. The results are compared and competed with each other based on their performance metrics. The characteristic values are presented in Fig. 8. It can be seen that the BR algorithm gives the best results. The LM algorithm leads to slightly inferior results. The SCG algorithm leads to the worst results. Furthermore, it has a comparatively high dispersion.

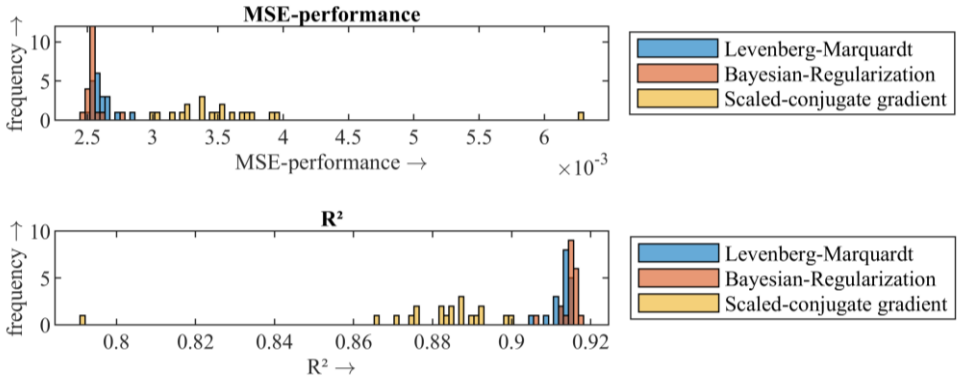


Fig. 8. Performance metrics with varying algorithms

A summary of further averaged parameters is shown in Tab. 2. The values highlighted in green are the best. The values highlighted in red are the worst.

Tab. 2. Performance metrics of varying algorithms

Performance metrics	LM algorithm	BR algorithm	SCG algorithm
Best training performance (preferably low)	0.0027	0.0027	0.0037
Best validation performance (preferably low)	0.0027	NaN	0.0037
Best testing performance (preferably low)	0.0027	0.0026	0.0037
No. of training epochs (preferably low)	246	476	284
R^2-value (preferable high)	0.91	0.91	0.87
Minimum gradient (preferably low)	1.48e-5	9.31e-8	5.04e-4
Training time [s] (preferably low)	242	515	279

It is observed that the BR algorithm is superior to the competing algorithms in terms of performance. Regarding the R^2 value, the LM algorithm is equivalent. In terms of computation time and number of training epochs, the LM algorithm is superior. The SCG algorithm is inferior regarding the performance and the R^2-value. The computation time and the number of raining algorithms is only intermediate.

From the investigation the recommendation can be derived to use the BR algorithm for high demands on the result quality. For time-critical applications, the use of the LM algorithm is recommended. No recommendation can be derived for the SCG algorithm.

4. RESULTS AND VALIDATION

The results are displayed using an ANN based efficiency map (see Fig. 9). By multiplying by the original maximum values, the normalized values can be converted back to the initial values. A high efficiency gradient can be seen in the lower speed and torque areas. Due to the relatively low resolution of the measurement data in this range, the information content is not sufficient to describe the efficiency there. Accordingly, the areas from 0 % to 5 % of the torque and speed are not mapped.

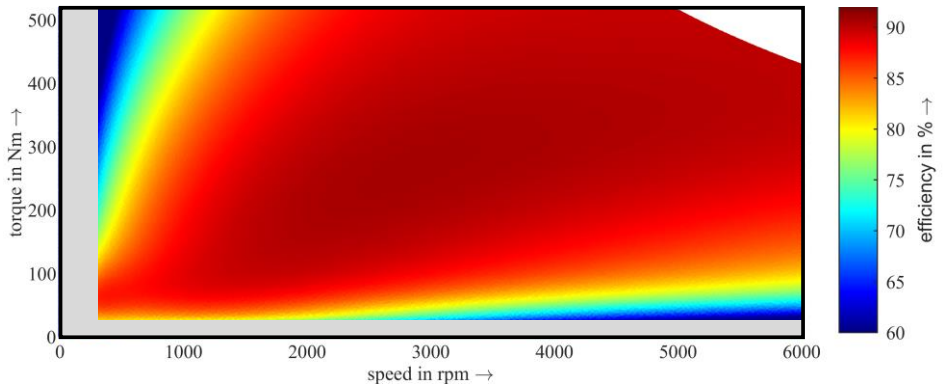


Fig. 9. Efficiency map derived from the ANN

Figure 10 shows a practical way of validation. For this purpose, Fig. 10 a shows the measured data in a specified time window. Fig. 10 b compares the time signal of the calculated and the measured efficiency. A high level of agreement is evident. However, the calculated value shows a smoother course than the measured value. This is attributable to residual oscillations in the measured current signal.

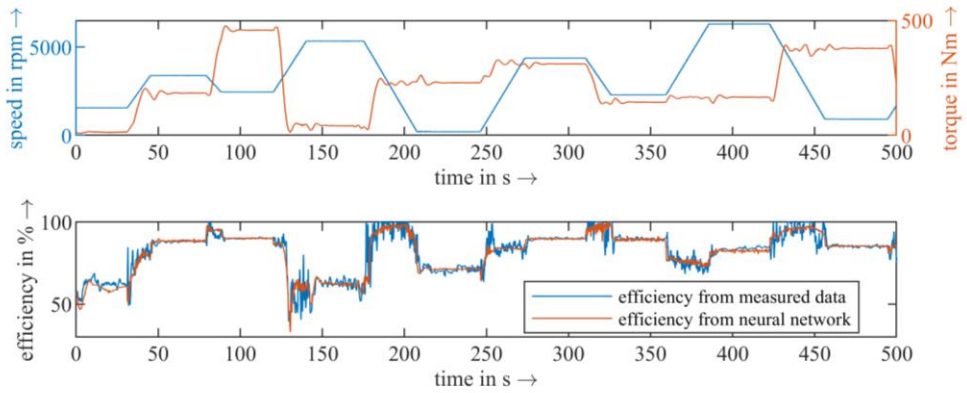


Fig. 10. Speed and torque corresponding to the efficiency from the measured data and the ANN

For a quantitative validation of the calculated values it is recommendable to define to a performance indicator. Therefore, the value x_η is introduced. It's calculated as the quotient of the efficiencies from the measured data η_{Calc} and from the ANN η_{NN} according to formula (8).

$$x_\eta = \frac{\eta_{Calc}}{\eta_{NN}} \tag{8}$$

where: x_η – efficiency quotient,

η_{Calc} – efficiency from measured data,

η_{NN} – efficiency from ANN calculation.

The variation in percentage between the efficiencies can be derived from the indicator. A value of $x_\eta = 1$ describes a perfectly fitted efficiency. A value of $x_\eta < 1$ describes a too high efficiency calculated by the ANN. A value off $x_\eta > 1$ describes a too low efficiency calculated by the ANN.

A representation of the indicator in the form of a histogram is shown in Fig. 11. It is evident that the calculated values are close to the measured values. For a numerical statement a statistical investigation is carried out. For this purpose, key parameters of the distribution, such as the standard deviation or the mean value, are calculated.

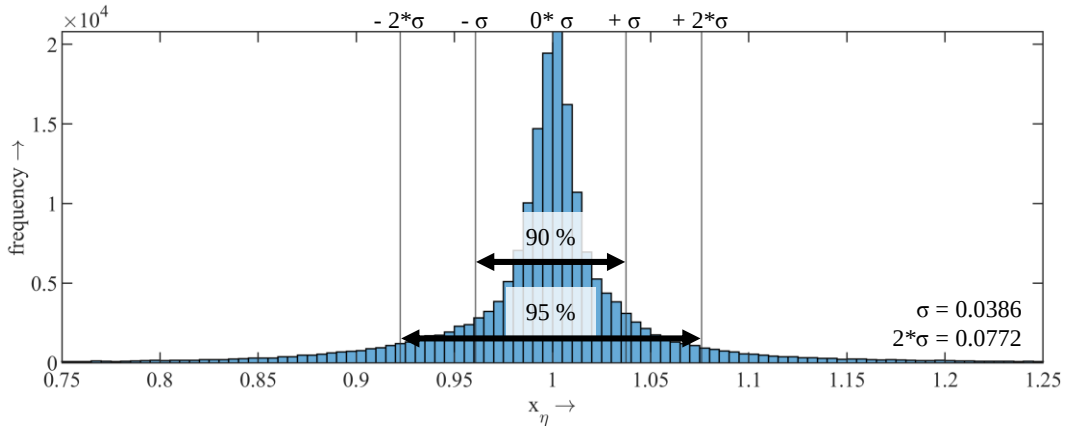


Fig. 11. Distribution and standard deviation of the performance indicator

The first standard deviation is σ = 3.86%. The second standard deviation is calculated as $2*\sigma$ = 7.72%. About 90% of the determined values for x_η are within the first standard deviation. About 95% of the determined values for x_η are within the second standard deviation. Thus, the dispersion of the values is classified as sufficiently low.

5. CONCLUSION

With the use of ANNs, the efficiency of a drive unit consisting of a DC motor and a planetary transmission could be described mathematically. The resulting function enables a variety of applications, such as the observation of the efficiency curve in real and simulated data or the determination and representation of characteristic efficiency maps. The application to preexisting models by integration as a subsystem enables an approximation of the models to reality. The informative value of the prediction increases. The time-efficient calculation by the determined function enables the integration into real-time simulation applications. The application to real measurement data enables the early detection of optimal operating points. This allows the operating strategy to be optimized.

The modelling approach specifically for efficiency can be transferred to other systems, such as separately considered transmissions, without a drive unit. The experimental modelling by ANNs can be transferred to other subsystems, such as the control system, in addition to the simulation of the efficiency.

The comprehensive description of the formation of the function also includes the design of experiments, data generation and data pre-processing. Especially problem solving in data preprocessing is presented with effective and efficient approaches.

The limitations of the approach are, for example, the inability to reliably extrapolate outside the measured data. In addition, the internal processes are not known due to the use of the black-box model.

For further optimization, potentials were identified. For example, the quality of the training data can be optimized by further optimization of the experimental design, in particular by an intelligent distribution of the measurement point density analogous to Martini et al. (2003).

REFERENCES

Albers, A., Behrendt, M., Klingler, S., & Matros, K. (2016). Verifikation und Validierung im Produktentstehungsprozess [E-Book]. In M. Behrendt, S. Klingler & K. Matros (Eds.), *Handbuch Produktentwicklung* (pp. 541–557). Carl Hanser Verlag. https://doi.org/10.3139/9783446445819.019

Bauer, L., Bauer, M., & Kley, M. (2021). Modelbasierte Validierung der Prüfstandsdynamik zur Erprobung von Komponenten elektrifizierter Antriebsstränge mithilfe eines digitalen Zwillings. *Stuttgarter Symposium für Produktentwicklung, SSP 2021*, 105–116. https://doi.org/10.18419/opus-11478

Bauer, L., Beck, P., Stütz, L., & Kley, M. (2021). Enhanced efficiency prediction of an electrified off-highway vehicle transmission utilizing machine learning methods. *Procedia Computer Science*, *192*, 417–426. https://doi.org/10.1016/j.procs.2021.08.043

Beine, M., & Rasche, R. (2018). Datenmanagement für das szenariobasierte Testen. *ATZextra*, *23*(S4), 20–25. https://doi.org/10.1007/s35778-018-0024-9

ÇElik, E., Gör, H., ÖZtürk, N., & Kurt, E. (2017). Application of artificial neural network to estimate power generation and efficiency of a new axial flux permanent magnet synchronous generator. *International Journal of Hydrogen Energy*, *42*(28), 17692–17699. https://doi.org/10.1016/j.ijhydene.2017.01.168

Dismon, H. (2017). Wir sind gefordert, Entwicklungen schnell und treffsicher umzusetzen. *MTZextra*, *22*(S1), 8–11. https://doi.org/10.1007/s41490-017-0009-4

Dohmen, H., Pfeiffer, K., & Schyr, C. (2009). *Antriebsstrangprüftechnik: Vom stationären Komponententest zum fahrmanöverbasierten Testen (Die Bibliothek der Technik (BT))* (1. Aufl.). Süddeutscher Verlag onpact.

German Environment Agency. (2020). *Submission under the United Nations Framework Convention on Climate Change and the Kyoto Protocol 2020*.

Guggenmos, J., Rückert, J., Thalmair, S., & Wagner, M. (2018). Das Prüffeld der Antriebsentwicklung im Wandel. *VPC – Simulation und Test 2015* (pp. 1–13). Springer. https://doi.org/10.1007/978-3-658-20736-6_1

Hoekstra, A. (2019). The Underestimated Potential of Battery Electric Vehicles to Reduce Emissions. *Joule*, *3*(6), 1412–1414. https://doi.org/10.1016/j.joule.2019.06.002

Isermann, R. (2007). *Mechatronische Systeme*. Springer.

Jazayeri, K., Jazayeri, M., & Uysal, S. (2016). Comparative Analysis of Levenberg-Marquardt and Bayesian Regularization Backpropagation Algorithms in Photovoltaic Power Estimation Using Artificial Neural Network. *Advances in Data Mining. Applications and Theoretical Aspects* (pp. 80–95). Springer. https://doi.org/10.1007/978-3-319-41561-1_7

Khan, A., Mohammadi, M. H., Ghorbanian, V., & Lowther, D. (2020). Efficiency Map Prediction of Motor Drives Using Deep Learning. *IEEE Transactions on Magnetics*, *56*(3), 1–4. https://doi.org/10.1109/tmag.2019.2957162

Li, Y. L., Kley, M., & Wang, S. J. (2014). Driveline Simulation of 2013 Formula Student Electric Racing Vehicle. *Applied Mechanics and Materials*, *541–542*, 424–429. https://doi.org/10.4028/www.scientific.net/amm.541-542.424

Machrowska, A., Karpiński, R., Jonak, J., & Krakowski, P. (2020). Numericalprediction of the component-ratio-dependent compressive strength of bone cement. *Applied Computer Science*, *16*(3), 88–101. https://doi.org/10.23743/acs-2020-24

Martini, E., Voß, H., Töpfer, S., & Isermann, R. (2003). Effiziente Motorapplikation mit lokal linearen neuronalen Netzen. *MTZ - Motortechnische Zeitschrift*, *64*(5), 406–413. https://doi.org/10.1007/bf03226705

Paulweber, M., & Lebert, K. (2014). *Mess- und Prüfstandstechnik: Antriebsstrangentwicklung Hybridisierung Elektrifizierung (Der Fahrzeugantrieb)* (2014. Aufl.). Springer. https://doi.org/10.1007/978-3-658-04453-4

Payal, A., Rai, C. S., & Reddy, B. V. R. (2013). Comparative analysis of Bayesian regularization and Levenberg-Marquardt training algorithm for localization in wireless sensor network. *15th International Conference on Advanced Communications Technology (ICACT)* (pp. 191–194). IEEE. https://ieeexplore.ieee.org/document/6488169

Ratov, D., & Lyfar, V. (2020). Modeling transmission mechanisms with determination of efficiency. *Applied Computer Science*, *16*(1), 33–40. https://doi.org/10.23743/acs-2020-03

Stütz, J., Bauer, L., & Kley, M. (2019). Intelligente Lastkollektivoptimierung für Erprobungen von elektrischen und hybriden Antriebssträngen. *Stuttgarter Symposium für Produktentwicklung SSP 2019* (pp. 93–102). Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO. https://doi.org/10.18419/opus-10394

Stütz, L., Beck, P., & Kley, M. (2021). Wirkungsgraduntersuchungen am Antriebsstrang von Multifunktionsfahrzeugen unter Berücksichtigung von empirisch ermittelten Lastkollektiven. *Stuttgarter Symposium für Produktentwicklung SSP 2021* (pp. 445–454). Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO. https://doi.org/10.18419/opus-11478

The MathWorks. (2020). *Statistics and Machine Learning Toolbox User's Guide*. The MathWorks.

Willmerding, G., & Häckh, J. (2017). Echtzeitsimulation hochdynamischer Fahrzeugantriebe. *ASIM-Treffen STS/GMMS 2017* (pp. 192–198). Ulm.

Yadav, R. N., & Yadava, V. (2017). Artificial neural network modelling of erosion-abrasion-based hybrid machining of aluminium-silicon carbide-boron carbide composite. *International Journal of Engineering Systems Modelling and Simulation*, *9*(2), 63–77. https://doi.org/10.1504/ijesms.2017.083223

Applied Computer Science, vol. 17, no. 4, pp. 20–33
doi: 10.23743/acs-2021-26

Submitted: 2021-10-21 | Revised: 2021-11-29 | Accepted: 2021-12-10

Keywords: hardware trojan, chips, logic test, machine learning, microcontroller

Kadeejah ABDULSALAM [0000-0001-7856-4269]*, *John ADEBISI* [0000-0002-1105-7219]*, *Victor DUROJAIYE**

IMPLEMENTATION OF A HARDWARE TROJAN CHIP DETECTOR MODEL USING ARDUINO MICROCONTROLLER

Abstract

These days, hardware devices and its associated activities are greatly impacted by threats amidst of various technologies. Hardware trojans are malicious modifications made to the circuitry of an integrated circuit, Exploiting such alterations and accessing the level of damage to devices is considered in this work. These trojans, when present in sensitive hardware system deployment, tends to have potential damage and infection to the system. This research builds a hardware trojan detector using machine learning techniques. The work uses a combination of logic testing and power side-channel analysis (SCA) coupled with machine learning for power traces. The model was trained, validated and tested using the acquired data, for 5 epochs. Preliminary logic tests were conducted on target hardware device as well as power SCA. The designed machine learning model was implemented using Arduino microcontroller and result showed that the hardware trojan detector identifies trojan chips with a reliable accuracy. The power consumption readings of the hardware characteristically start at 1035-1040mW and the power time-series data were simulated using DC power measurements mixed with additive white Gaussian noise (AWGN) with different standard deviations. The model achieves accuracy, precision and accurate recall values. Setting the threshold probability for the trojan class less than 0.5 however increases the recall, which is the most important metric for overall accuracy acheivement of over 95 percent after several epochs of training.

1. BACKGROUND

Hardware Trojan (HT) is a malicious modification of the circuitry of an integrated circuit (IC) (Salmani, Tehranipoo & Plusquellic, 2011). Most hardware trojans are specially designed to change the functionality, reduce the reliability and/or leak valuable information from the host circuit. An HT's payload is the whole activity the trojan performs when triggered. Malicious trojans generally attempt to bypass or disable a system's safety arcade; and private data can be leaked by radio emission. HTs could also disable, derange or ruin the whole chip

* University of Lagos, Electrical Electronics and Computer Engineering Department, Nigeria, kabdulsalam@unilag.edu.ng, adebisi_tunji@yahoo.com, durojaiyevic@gmail.com

or its parts (Salmani, Tehranipoo & Plusquellic, 2011). Albeit trojans are designed for various purposes, they can be functional or parametric, small or large, tight or loose, logic or sensor-based, trigger based or always on among others. Different techniques can be used for trojan detection, each with its strengths and weaknesses. Some of the weaknesses could be attached to the test nature (Grus, 2015); such as – destructive tests, which involve removing the chip from the package and scanning the layers to get the net lists, possibly using an electron microscope (Grus, 2015). The floor plans gotten from the process can then be compared with the floor plans of the actual IC.

This involves reverse-engineering (He et al., 2014). However, this process renders the chip useless, and the presence (or absence) of trojans in a chip does not guarantee the presence (or absence) in another chip (Grus, 2015). In this case, non-destructive tests are alternatives which do not involve the reverse-engineering of the chip. Other tests include; Logic test where the input ports of the chip are stimulated with test vectors, and the output is compared to what is expected. A deviation from the expected output could indicate the existence of a trojan (Wang & Luo, 2011). Power SCA; observes power traces from the device under test (DUT) and compares the results to what is obtained from a "golden chip" (a trusted chip without any trojans). The existence of a large trojan would imply an obvious difference in power consumption. However, for smaller trojans, more sophisticated approaches involving machine learning (ML) would need to be used to identify the subtle differences between chips which contain trojans and those which do not (Ni et al., 2014). Delay SCA: uses a time delay in the path of DUT, which is more than the time delay observed in a golden chip. It could indicate the presence of a trojan in a path while runtime test monitors a running system, actively checking for indications that a trojan may exist, but contains an interrupt mechanism to stop the system once a trojan is detected, protecting the system (Cui et al., 2016). A summary of the different types of hardware trojans and the detection methods is presented in Table 1.

Tab. 1. Summary of hardware trojan detection methods

Trojans	Logic test	Power SCA	Delay SCA	Run time
Parametric	×	✓	✓	×
Big	?	✓	?	✓
Small	✓	×	✓	?
Tight	✓	✓	✓	?
Loose	✓	?	✓	?
Always-on	×	×	✓	×
Leak info	×	✓	×	✓

where: ✓ – good, × – bad, **?** – depends.

2. LITERATURE REVIEW

In literature, hardware trojan is a deliberate and unwanted alteration of an integrated circuit (Jahan, Sajal & Nygard, 2019; Salmani, Tehranipoo & Plusquellic, 2011). They present emerging security concerns and can have devastating effects when used in sensitive

environments. Hence, research of this nature is needed to detect them. Physical inspection could be destructive and sometimes involves reverse-engineering which makes the chip unusable, the results of this test do not bring a lot of confidence; the presence (or absence) of a trojan is not guaranteed and such test is not standard enough to be used in a dynamic environment (Paul, Suman & Sultan, 2013).

Research has been done on the detection of hardware trojans using machine learning involving Principal Component Analysis (PCA), Latent Dirichlet Analysis (LDA) or Support Vector Machine (SVM). In (He et al., 2014), a combination of logic testing and side-channel testing was used as a novel hardware trojan detection method. An 18-bit Intellectual Property (IP) core was used as a golden circuit, while a 2-bit counter was used as a trojan circuit. Testing was done using the Xilinx ISE (Integrated Synthesis Environment), LabVIEW software and a high-precision oscilloscope. The power traces from the devices were monitored and PCA was used to process them, extracting three projections on three corresponding largest eigenvectors. In the resulting 3D graph, there was a clear difference between the devices as the power traces for each class clustered together. The system was shown to have a trojan detection sensitivity of 0.1%.

The use of machine learning for power SCA was explored in (Shende & Ambawade, 2016). Here, the experimental setup involves a golden chip; an AES (Advanced Encryption Standard) core, and another with a trojan inserted. The hardware is synthesized with using Xilinx ISE and implemented on a Xilinx Spartan-6 Field Programmable Gate Array (FPGA). Power measurements were obtained using power analysis exploratory data analysis (EDA) tools, dimensionality reduction is done using PCA and then classified using LDA. The setup is able to differentiate between the golden and trojan devices to a very high degree of accuracy, theoretically 100%.

A research on "detection technique for hardware trojans using machine learning in frequency domain", which is an application of SCA on the power consumption waveform data limited to the frequency domain using discrete Fourier transform (DFT) of the waveforms is presented in (Iwase, Nozaki, Yoshikawa & Kumaki, 2015). An AES core and several other variants containing trojans were built on an FPGA using Verilog. The Fourier transform of the power traces is then used to train an SVM machine learning model with some level of accuracy recorded. The work of (Bai, 2018; Cui et al., 2016) focused more on Cluster Analysis of Mahalanobis Distance in their detection approach; a concept associated with LDA. An AES encryption circuit was designed on an FPGA, and on another design, a trojan is added. Power SCA was then carried out. Results from the experiments show that in performing cluster analysis on the data points, the Mahalanobis distance gives far more accurate results than the Euclidean distance.

The Influence on Sensitivity of Hardware Trojans Detection by Test Vector" was investigated in (Ni et al., 2014), although the research was a proposal of power relative variation (PRV) parameters to analyze the relation between test vectors and detection sensitivity. S-box circuit noise model was introduced to an AES core and the power traces of various sizes of trojans were simulated using HSPICE. Different sets of test vectors are applied as well. It was found that the power of the entire circuit is reduced by 41% and the PRV value increased 12.71% and 3.34% at most, corresponding to combinational and sequential trojan circuits. A power characteristic template for hardware trojan detection which is a novel hardware detection method using PCA and applying the Mahalanobis distance was developed in (Zhang et al., 2016). The output could adapt to a variety of different

functions on the same chip. The RS232 serial port hardware trojan implanted in DES and AES algorithms was used for verification. Hardware trojan detection rate was high, with low computational cost due to dimensionality reduction. In the work of (Wang & Luo, 2011), a power analysis based experimental approach was used coupled with two FPGAs; one used for implementing the 64-bit DES cipher (called "Test FPGA") and the other used to generate test vectors (called "FPGA pattern generator"). Two trojans were designed for the DES cipher and the power traces were analyzed through singular value decomposition (SVD). This detection method works, even for trojans about two orders of magnitude smaller than the main circuit.

A General Framework for Hardware Trojan Detection in Digital Circuits by Statistical Learning Algorithms" – presented in (Chen et al., 2016), uses a Bayesian inference-based calibration technique to check for the existence of a trojan and map to the sparse solution of the linear system A batch of underdetermined linear systems are solved together by the well-studied simultaneous orthogonal matching pursuit algorithm to get their common sparse solution. This framework gives high trojan detection rates with low measurement cost. It has been observed power SCA is the leading method of identifying hardware trojans in literature. This research leverage on the work, along with the concept of logic testing. Machine learning was used to create a smart and accurate trojan detection system. It was observed that the use of deep learning techniques would provide better results when working with time-series data (power traces of chips), compared to more traditional machine learning approaches such as PCA, LDA and SVM. Hence, deep learning would be used for power SCA, more specifically a mix of convolutional neural networks (CNN) and recurrent neural networks (RNN).

3. DESIGN METHODOLOGY

This section contains the procedures and methods used in building the proposed trojan detector. It further, highlights the software and hardware tools used for model implementation. Theoretical frameworks for the different components and subsystems involved in the design are also considered in this section.

3.1. Field programmable gate array (FPGA)

This is a programmable hardware used to implement any logic circuit; combinational or sequential. It serves as an alternative to designing an application-specific integrated circuit (ASIC). FPGAs contain programmable logic blocks, programmable interconnects and input/output (I/O) blocks as shown in Figure 1.

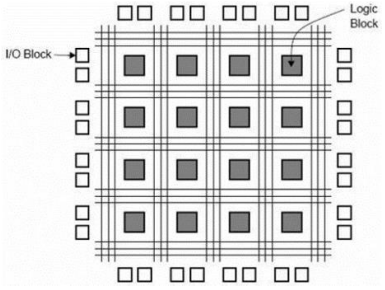


Fig. 1. Block diagram of an FPGA [source: (Tutorial Point, 2020)]

Several hardware description languages (HDLs) can be used for programming FPGAs. The most popular ones are Verilog and VHDL (Very High Speed IC Hardware Description Language). For this research, VHDL was used due to its deterministic and better error-checking capabilities. Architectures can be modelled using a structural description (which shows the interconnections of components), a behavioural description (a high-level description of how the circuit should behave) or a combination of both. In this paper a Cyclone II EP2C5 Mini Development Board was used for the hardware design. It has 4608 logic elements (16 logic elements per block), 26kb memory blocks, 13 embedded multipliers, 2 phase-locked loops (PLLs) and 89 usable I/O ports. Its compact size, low cost and high efficiency makes it a good choice for the design.

3.2. The chip model design

The base chip model designed is a synchronous 10-bit binary counter, which counts the number of occurrences of a trigger signal (clock), incrementing the count whenever a clock pulse occurs. The n-bit counter used contains n flip-flops which can hold values from 0 to $2^n - 1$, where n is the number of bits used for the counter. Hence, the 10-bit counter designed counts from 0 (0000000000) to 1023 (1111111111). Typically, once counters reach maximum count, go back to 0 and start all over. However the counter module of this research would be triggered by a positive-going clock transition, from 0 to 1 while, an asynchronous reset/clear input would be present, resetting the count to 0 whenever triggered. The elaborated design for the golden chip is shown in Figure 2. The VHDL source can be seen in Figure 3.

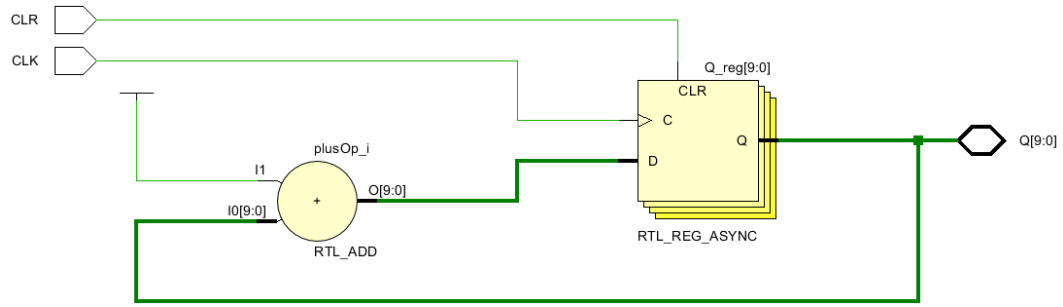


Fig. 2. Golden chip design

Given that the clock period is 10ms, the time it takes to complete a data gathering round is approximately 10 ms; 1024 = 10240 ms = 10.24 seconds. The trojan chip shares a lot of similarities with the golden chip, as expected for a device with a trojan. However, for the trojan device, count 512 (1000000000) is replaced with 1023 (1111111111) and it remains in this state until reset. This acts as a malicious modification to the base circuit. The trojan chip design is shown in Figure 4. The VHDL source can be seen in Figure 5.

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity golden_chip is
    Port ( CLK : in STD_LOGIC := '0';
           CLR : in STD_LOGIC := '0';
           Q : inout STD_LOGIC_VECTOR (9 downto 0) := (others => '0'));
end golden_chip;

architecture Behavioral of golden_chip is
begin
    process(CLK, CLR)
    begin
        if CLR = '1' then
            Q <= (others => '0');
        elsif rising_edge(CLK) then
            Q <= Q + 1;
        end if;
    end process;

end Behavioral;
```

Fig. 3. Golden chip VHDL code

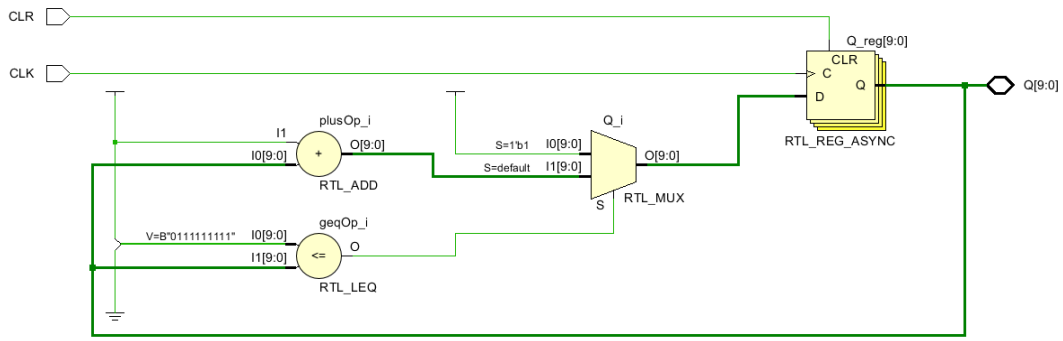


Fig. 4. Trojan chip design

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity trojan_chip is
    Port ( CLK : in STD_LOGIC := '0';
           CLR : in STD_LOGIC := '0';
           Q : inout STD_LOGIC_VECTOR (9 downto 0) := (others => '0'));
end trojan_chip;

architecture Behavioral of trojan_chip is
begin
    process(CLK, CLR)
    begin
        if CLR = '1' then
            Q <= (others => '0');
        elsif rising_edge(CLK) then
            if Q >= "0111111111" then
                Q <= (others => '1');
            else
                Q <= Q + 1;
            end if;
        end if;
    end process;

end Behavioral;
```

Fig. 5. Trojan chip VHDL code

3.3. Data acquisition with Arduino microcontroller

In this research, data used for the trojan detection was acquired using an Arduino microcontroller due to its flexibility and durability, albeit in previous works, LabVIEW was used for data acquisition, where data was read off the chip and sent to the computer for processing. Arduinos typically have a USB port where they can be interfaced with a computer for programming, and also for serial communication. The Arduino Uno used for this research addressed power measurements issues for SCA purposes, in addition, a INA219 current sensor was used for both voltage and current data acquisition and measurements, thus, the power computed by multiplying the instantaneous voltage and current readings. Figures 6 and 7 show the printData function and the microcontroller setup.

A python script was used for the extraction of the data at the other end of the serial COM port; pushed into the COM port by the Arduino sketch (the data acquisition system). The number of rounds is first specified as an input to the script, determining the amount of data to be gathered from the chip. The type of chip is also specified as an input, be it a golden or a trojan chip, so as to store the dataset in the appropriate location. Once the parameters have been determined, the script first resets the counter by communicating with the Arduino writing a custom 'RESET' command to the COM port. Afterwards, data is then collected a number of times, as specified by the input parameters, saving the data in .csv form.

```
void printData(float t) {
  int Q9 = digitalRead(2);
  int Q8 = digitalRead(3);
  int Q7 = digitalRead(4);
  int Q6 = digitalRead(5);
  int Q5 = digitalRead(6);
  int Q4 = digitalRead(7);
  int Q3 = digitalRead(8);
  int Q2 = digitalRead(9);
  int Q1 = digitalRead(10);
  int Q0 = digitalRead(11);

  float shuntvoltage = ina219.getShuntVoltage_mV();
  float busvoltage = ina219.getBusVoltage_V();
  float current_mA = ina219.getCurrent_mA();
  current_mA = abs(current_mA);
  float loadvoltage = busvoltage + (shuntvoltage / 1000);
  float power = current_mA * loadvoltage;

  Serial.println((String) t + "," +
                 (String) Q9 +
                 (String) Q8 +
                 (String) Q7 +
                 (String) Q6 +
                 (String) Q5 +
                 (String) Q4 +
                 (String) Q3 +
                 (String) Q2 +
                 (String) Q1 +
                 (String) Q0 + "," +
                 (String) power);
}
```

Fig. 6. An example of printData function

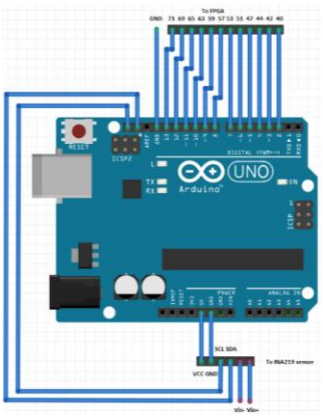


Fig. 7. Arduino data acquisition system

3.4. The Machine Learning model

A binary classification machine learning model which processes time-series data (power traces) and classifies each time-series data was built for the logic testing. It provides insight into whether or not the chip of interest contains a trojan. This is an hardware trojan detection technique which involves running test vectors through the input, checking for discrepancies at the output of the chip. A discrepancy thus indicate the presence of a trojan. If a chip fails the logic test; there is no need to proceed any further; it is tagged as containing a trojan. However, more sophisticated trojans, especially trigger-based and parametric trojans, may pass the logic testing stage. Nonetheless, a more sophisticated detection method would be required. This test result was used in this case to generate training data for the machine learning model. For both the golden and trojan chips, logic test was carried out severally and the power traces obtained were written into files, for later use in training the model. One hundred iterations were done to obtain large data in training the model for accuracy. However, there is a tradeoff between the accuracy of the model and how long it takes the model to train, as more training data increases the training time of the model.

The power SCA hardware trojan detection methodology was employed through monitoring of the power consumption (power trace) of the chip and analyzing the time-series to find patterns and detect whether or not a trojan is present in the device of interest. The power trace is monitored during logic testing, in anticipation of the chip passing the test. Once this happens, power SCA would be used to get more insight into the trojan status of the chip. This involves a sophisticated machine learning model which has been pre-trained. The power trace for any chip is bound to be different every time due to stochastic variations. However, there are underlying patterns in the power consumption for the chips, which is difficult for the human eye to observe, but easy for a sensitive machine learning model to discern. An elegant solution used in this research was a collection of values evenly spaced in time. These values, structured as one-dimensional arrays of length n, can be seen as representing a point in n-dimensional space. The time-series was then processed by the machine learning algorithm in this form; as a single data point. This is not far-fetched from the fact that deep learning approaches was used. The trained model is a neural network. See Figures 8 and 9 for the architecture of the neural network and the code which creates this model.

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1024, 32)          64
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 1024, 32)          204832
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 512, 32)           0
_________________________________________________________________
dropout_1 (Dropout)          (None, 512, 32)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               53200
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 101
=================================================================
Total params: 258,197
Trainable params: 258,197
Non-trainable params: 0
```

Fig. 8. Neural network architecture

```
# create the model
model = Sequential()
model.add(Dense(32, input_shape=(N,1)))
model.add(Conv1D(filters=32, kernel_size=200, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.1))
model.add(LSTM(100))
model.add(Dropout(0.1))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

Fig. 9. Neural network code

4. MODEL IMPLEMENTATION

The software tools used for this research work are: **Altera Quartus II** (for programming the Cyclone II EP2C5 FPGA), **Arduino IDE** (where the sketches were written, compiled and loaded to the Arduino) and **IDLE** (the official IDE for Python). Incremental software development model was used due to its flexibility and the requirements nature of the entire research which are clear and specific.

4.1. Trojan detecting procedure

A comprehensive summary of the trojan detection procedure is illustrated in Figure 10, as a flow chart.

The implementation was setup on GitHub – Version Control System (VCS) which allows developers to track changes in code. During development, the different states of the Arduino, VHDL and Python source codes at various points were organised in snapshots called commits. This proved useful in a number of cases where changes which broke a feature had to be removed by rolling back to a previous commit or getting a few lines of code which existed in an older version of the source file of interest. GitHub served as a remote/online version of the model implementation.

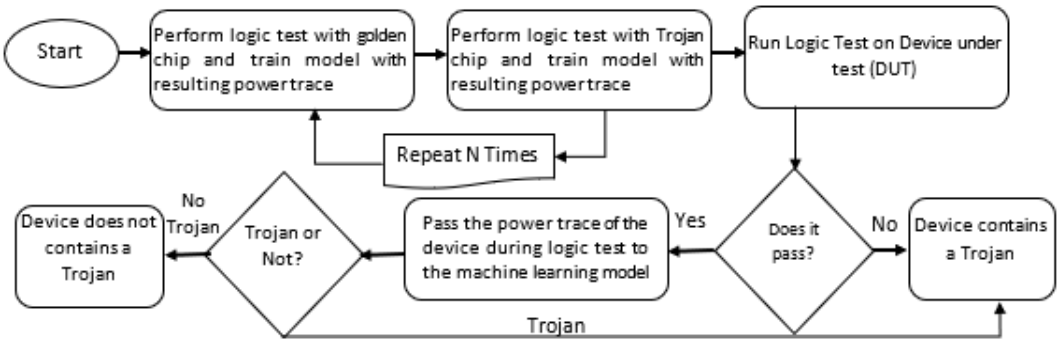


Fig. 10. Trojan detection procedure

4.2. Training the model

The model is a neural network which contains convolutional layers from CNNs and feedback layers as seen in RNNs, or more specifically, long short-term memory (LSTM) networks. A Python library Keras was used for building the network using Scikit-learn as a scale for the data and compute the model metrics (accuracy, precision and recall). At this stage all the data stored in text files are extracted with the aid of load_data and load_series python functions. See Figure 11 for code snippet.

```
def load_series(folder, file):
    t = []
    series = []
    with open(join(folder, file)) as f:
        data = f.read().split('\n')
        for line in data:
            linearr = line.split(',')
            t.append(float(linearr[0]))
            series.append(float(linearr[2]))
    return t, series
```

Fig. 11. An example of 'load_series' function

Afterwards, the data was normalized and the model trained on the training dataset. Following the training, and device testing using power SCA (i.e. after passing the logic test), the time-series is then converted to a data point and passed as input to the trained neural network. The neural network outputs a probability; the probability that the chip contains a trojan. Based on a predefined threshold, the probability is converted to a binary decision afterwards; whether or not the chip contains a trojan. Testing was then done using the testing dataset to determine the accuracy, precision and recall of the model.

4.3. Discussion of result

The data acquisition system was able to collect counter state data and power consumption data for both the golden and trojan chips. The data was structured in CSV using the format "time (in seconds), counter state, power consumption (in mW)". Figures 12 and 13 show some data samples for the golden and trojan chips respectively. Observe that, as expected, the trojan chip deliberately gives the wrong state for counts after 511.

```
5.00,0111110100,1026.32
5.01,0111110101,1026.33
5.02,0111110110,1023.51
5.03,0111110111,1025.32
5.04,0111111000,1024.50
5.05,0111111001,1024.83
5.06,0111111010,1026.00
5.07,0111111011,1026.82
5.08,0111111100,1025.32
5.09,0111111101,1026.32
5.10,0111111110,1025.33
5.11,0111111111,1023.51
5.12,1000000000,1024.84
5.13,1000000001,1025.65
5.14,1000000010,1024.50
5.15,1000000011,1026.82
5.16,1000000100,1025.50
5.17,1000000101,1026.50
5.18,1000000110,1025.82
5.19,1000000111,1026.32
5.20,1000001000,1024.33
```

Fig. 12. Some data samples for the golden chip

```
5.00,0111110100,1020.54
5.01,0111110101,1021.04
5.02,0111110110,1020.71
5.03,0111110111,1020.21
5.04,0111111000,1021.04
5.05,0111111001,1021.04
5.06,0111111010,1021.54
5.07,0111111011,1021.54
5.08,0111111100,1020.54
5.09,0111111101,1021.04
5.10,0111111110,1020.05
5.11,0111111111,1021.54
5.12,1111111111,1021.54
5.13,1111111111,1020.04
5.14,1111111111,1023.02
5.15,1111111111,1020.04
5.16,1111111111,1021.04
5.17,1111111111,1020.04
5.18,1111111111,1021.54
5.19,1111111111,1022.03
5.20,1111111111,1020.54
```

Fig. 13. Some data samples for the trojan chip

The model was trained, validated and tested using the acquired data, for 5 epochs. As expected, the accuracy increased and the model loss decreased. The validation and testing set also performed well on the data. More training epochs are possible but there would be an increased risk of overfitting the model. Figures 14 and 15 show graphs for the model performance over training epochs.

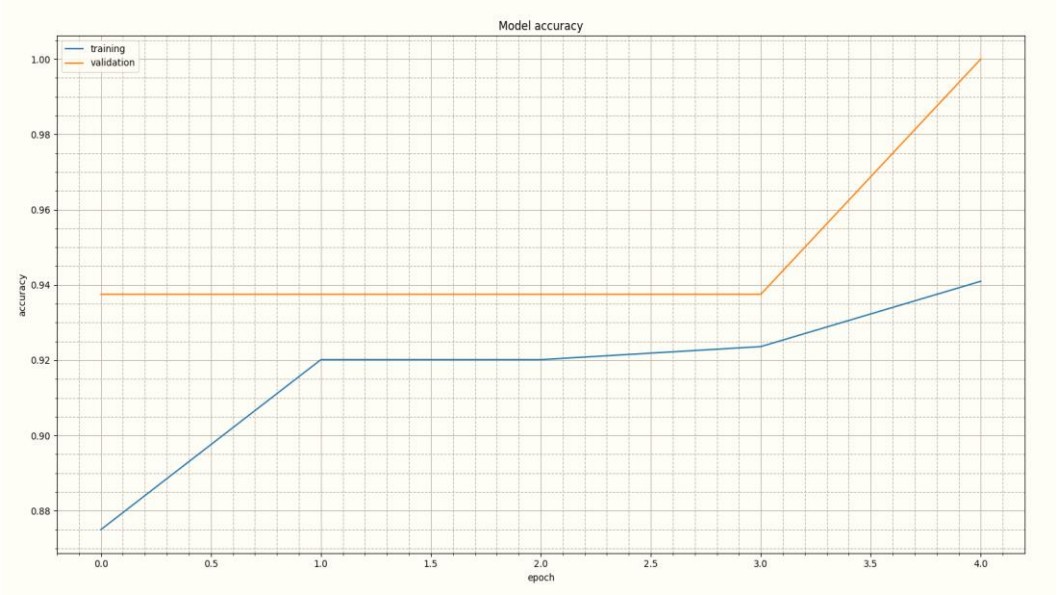
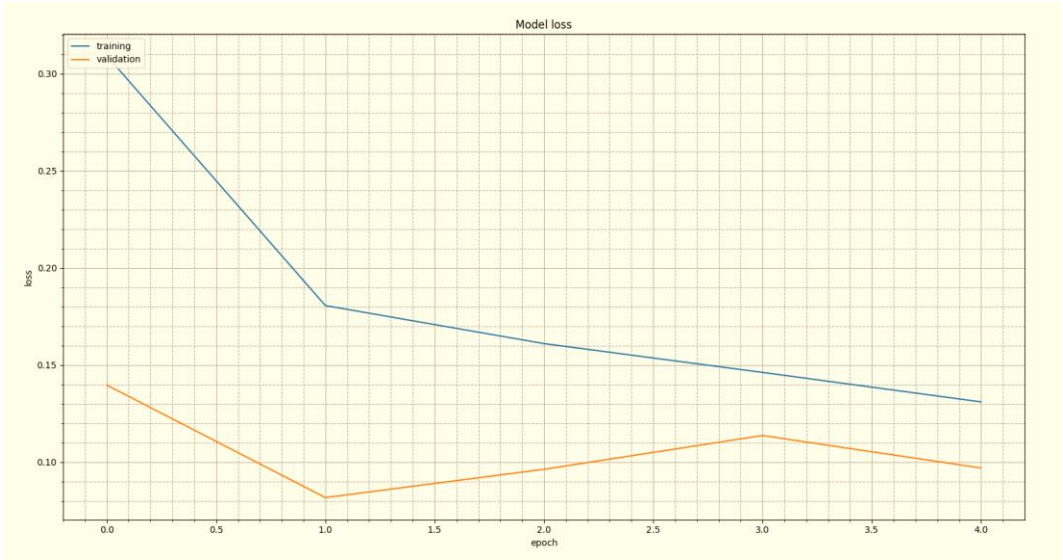


Fig. 14. Model accuracy



Fig, 15. Model loss

Following the successful training of the model, the system as a whole was tested using golden and trojan chips. The results are shown in Figures 16 and 17 respectively. Observe that both logic and power SCA tests give the same result.

```
Connected to: COM12
Test running...
Logic test: Passed

Trojan probability: 0.091
Power SCA test: Passed
>>>
```

Fig. 16. Testing a golden chip

```
Connected to: COM12
Test running...
Logic test: Failed

Trojan probability: 0.898
Power SCA test: Failed
>>>
```

Fig. 17. Testing a trojan chip

As depicted in the results, the trojan detection system works to a high degree of accuracy. The power consumption readings of the hardware characteristically start at 1035–1040mW. The power Time-series data were simulated by using DC power measurements mixed with additive white Gaussian noise (AWGN) with different standard deviations. The model was then trained and tested based on the simulated data. The recall improved by adjusting the threshold variable, although at the expense of too many false positives in the model's prediction. At the early stage, the model achieves accuracy, precision and recall values

of 80 to 90 percent. Setting the threshold probability for the trojan class less than 0.5 however increases the recall, which is the most important metric for the system later achieved overall accuracy, precision and recall values of over 95 percent after several epochs of training.

However, it should be noted that the detection system built was trained with specific implementation of a trojan; that alters the count of the counter in a very unique way. To kin future, the model could be subjected to varieties of trojan implementations for the base chip. The more trojan implementations the model learns from, the more accurately it can identify patterns in golden and trojan chips, and correctly classify them. Also, although the neural network works reasonably well, its architecture was mostly arbitrary. A possible follow-up research work is finding the optimal network architecture for training the model. Search techniques such as grid search, or optimisation techniques with a touch of genetic algorithms can also be applied to find the optimal hyperparameters for training the model.

5. CONCLUSION

Two chips – one golden and the other trojan were designed – both having hardware and software interface. A machine learning model was then built in Python and trained with data gathered from the hardware. This project illustrates a concept which solves the problem stated earlier in this paper. Using sophisticated and cutting-edge machine learning techniques, a system which can detect modification to integrated circuit designs has been built. This work provided a different and better approach to such malicious modifications to hardware considering the emerging security concerns. While novel methods are being created to combat them, the more recent trojans are intelligently written and are capable of evading detection by most methods. This work takes a more sophisticated approach in detecting these trojans, using machine learning.

REFERENCES

Bai, X. (2018). Text classification based on LSTM and attention. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)* (pp. 29–32). IEEE. https://doi.org/10.1109/ICDIM.2018.8847061

Chen, X., Wang, L., Wang, Y., Liu, Y., & Yang, H. (2016). A general framework for hardware trojan detection in digital circuits by statistical learning algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (vol. 36, no. 10, pp. 1633–1646). IEEE. https://doi.org/10.1109/TCAD.2016.2638442

Cui, Q., Sun, K., Wang, S., Zhang, L., & Li, D. (2016). Hardware trojan detection based on cluster analysis of mahalanobis distance. *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (pp. 234–238). IEEE. https://doi.org/10.1109/IHMSC.2016.65

Grus, J. (2015). *Data Science from Scratch. 1005 Gravenstein Highway North*. O'Reilly Media.

He, C., Hou, B., Wang, L., En, Y., & Xie, S. (2014). A novel hardware Trojan detection method based on side-channel analysis and PCA algorithm. *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)* (pp. 1043–1046). IEEE. https://doi.org/10.1109/ICRMS.2014.7107362

Iwase, T., Nozaki, Y., Yoshikawa, M., & Kumaki, T. (2015). Detection technique for hardware Trojans using machine learning in frequency domain. *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)* (pp. 185–186). IEEE. https://doi.org/10.1109/GCCE.2015.7398569

Jahan, I., Sajal, S. Z., & Nygard, K. E. (2019). Prediction model using recurrent neural networks. *2019 IEEE International Conference on Electro Information Technology (EIT)* (pp. 1–6) IEEE. https://doi.org/10.1109/EIT.2019.8834336

Ni, L., Li, S., Chen, J., Wei, P., & Zhao, Z. (2014). The influence on sensitivity of hardware trojans detection by test vector. *2014 Communications Security Conference (CSC 2014)* (pp. 1–6). IEEE. https://doi.org/10.1049/cp.2014.0756

Paul, L. C., Suman, A. A., & Sultan, N. (2013). Methodological analysis of principal component analysis (PCA) method. *International Journal of Computational Engineering & Management*, *16*(2), 32–38.

Tutorial Point. (2020). Retrieved October 8, 2021 from https://www.tutorialspoint.com

Salmani, H., Tehranipoor, M., & Plusquellic, J. (2011). A novel technique for improving hardware trojan detection and reducing trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *20*(1), 112–125. https://doi.org/10.1109/TVLSI.2010.2093547

Shende, R., & Ambawade, D. D. (2016). A side channel based power analysis technique for hardware trojan detection using statistical learning approach. *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)* (pp. 1–4). IEEE. https://doi.org/10.1109/WOCN.2016.7759894

Wang, L.-W., & Luo, H.-W. (2011). A power analysis based approach to detect Trojan circuits. *2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering* (pp. 380–384). IEEE. https://doi.org/10.1109/ICQR2MSE.2011.5976635

Zhang, L., Sun, K., Cui, Q., Wang, S., Li, X., & Di, J. (2016). Multi adaptive hardware Trojan detection method based on power characteristics template. *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)* (pp. 414–418). IEEE. https://doi.org/10.1109/CCIS.2016.7790294

Applied Computer Science, vol. 17, no. 4, pp. 34–44
doi: 10.23743/acs-2021-27

Submitted: 2021-07-22 | Revised: 2021-09-07 | Accepted: 2021-11-09

Keywords: demand forecasting, artificial neural network, price, promotion, federal funds rate

Anupa ARACHCHIGE[*], *Ranil SUGATHADASA*[*], *Oshadhi HERATH*[*], *Amila THIBBOTUWAWA* [0000-0002-5443-8839][*]

ARTIFICIAL NEURAL NETWORK BASED DEMAND FORECASTING INTEGRATED WITH FEDERAL FUNDS RATE

Abstract

Adverse effects of inaccurate demand forecasts; stockouts, overstocks, customer loss have led academia and the business world towards accurate demand forecasting methods. Artificial Neural Network (ANN) is capable of highly accurate forecasts integrated with many variables. The use of Price and Promotion variables have increased the accuracy while the addition of other relevant variables would decrease the occurrences of errors. The use of the Federal Funds Rate as an additional macro-economic variable to ANN forecasting models has been discussed in this research by the means of the accuracy measuring method: Average Relative Mean Absolute Error.

1. INTRODUCTION

Demand forecasting, projection of future demand for a specific product is a principal element for continuous balancing of demand and supply of that product (Hewage, Perera & De Baets, 2021). It is a major concern across all supply chains since most functions depend on demand forecasts and is a critical aspect of managing operations, procurement, production, local distribution, replenishment plans, transportation, sales, finance and marketing (Fildes, Ma & Kolassa, 2019; Parker, 2014; Oliva & Watson, 2009). Stockouts and overstocks are caused due to inaccuracy of demand forecasts which result in long-term customer dissatisfaction, customer loss, high inventory costs and waste of resources (Huang, Fildes, & Soopramanien, 2019; Yang, Goh, Xu, Zhang, & Akcan, 2015). Thus, accurate demand forecasting is crucial to take proactive measures in supply chain risk management (Perera, Thibbotuwawa, Rajasooriyar & Sugathadasa, 2016; Sugathadasa, Wakkumbura, Perera & Thibbotuwawa, 2021).

The foremost difficulty for accurate demand forecasts is the volatility due to unpredictable customer behaviour caused by endogenous and exogenous factors. Among these factors, sales promotions have a large impact on consumer behaviour causing changes in market

[*] University of Moratuwa, Faculty of Engineering, Department of Transport and Logistics, Moratuwa, Sri Lanka, anupa13@hotmail.com, ranils@uom.lk, oshadhik@uom.lk, amilat@uom.lk

demand and sales trends (Balachandra, Perera & Thibbotuwawa, 2020). In addition, macroeconomic variables linked with consumer behaviour influence the demand forecasts (Tangjitprom, 2012). Hence a single traditional statistical forecasting technique comprises historical sales data which is inadequate to deliver proper forecasts where the impact of related sales data to mitigate the bullwhip effect in the supply chain is proven (Matharage, Hewage & Perera, 2020). With the prerequisite of a method to incorporate many variables with significantly improving data availability, the capability of ANN which is a part of Artificial Intelligence which creates sales forecasts with high accuracy integrated with many variables is to be assessed with the integration of macroeconomic variables based on model's accuracy.

2. LITERATURE REVIEW

Several complex statistical methods and simple practical methods are found to be applied in time series demand predictions where Exponential Smoothing Models, Auto-Regressive Integrated Moving Average (ARIMA) Models, State Space and Structural Models, Nonlinear Models and Long Short Memory Models (LSTM) have been identified as the methods of forecasting. ML algorithms are exercised as an alternative to traditional methods in the recent past. It enables systems capability of automatic learning and improving from experience without being programmed or any human intervention. ML models are trained with a portion around 80% of the available historical data set and exercising the rest, the training dataset to evaluate the expected performance of the models (Harris, Nadler & Bhan, 1984; Ni, Xiao & Lim, 2019). As per literature, ML models are more accurate than the traditional demand forecasting methods especially Neural Networks (NNs) being capable of using non-linear algorithms in statistical predictions (Barker, 2020; Spiliotis, Makridakis, Semenoglou & Assimakopoulos, 2020). Thus, these functional improvements in forecasting models have improved the competitiveness of supply chains (Perera & Sudusinghe, 2017; Ranil, Sugathadasa, Senadheera & Thibbotuwawa, 2021).

Distinguishing the relative impact of various factors affecting the demand has been challenging in demand forecasting hence researchers have focused on incorporating such variables in forecasting models to improve its accuracy (Huang, Fildes & Soopramanien, 2014). The input variables are selected based on the importance of the variable with its impact relevancy towards the model (Abolghasemi, Eshragh, Hurley & Fahimnia, 2020). Thus, a superior forecasting performance with high accuracy can be achieved through the incorporation of competitive information by choosing the correct variables out of many available variables.

Price and promotion variables, being marketing tools in the form of price discounts affecting the sales level are highly incorporated in forecasting models with proven accuracy increment over baseline model (Guidolin, Guseo & Mortarino, 2019; Huang, Fildes & Soopramanien, 2014; Ma, Fildes & Huang, 2016). Studies have demonstrated the average of Pearson's Correlation between price and demand gained as -0.83 indicating a strong negative relationship which hypothetically explains a high percentage of variation in demand. The impact of promotions on sales forecasting is explicitly addressed by Ali et al. (2009) with time-series autoregressive models.

Moreover, the literature suggests the integration of macroeconomic variables to increase the accuracy levels in demand forecasting approaches. Macroeconomic variables such as unemployment, employment, inflation, Gross Domestic Product, interest rates are indicators

of economic performances of markets which can be used to increase the forecast accuracy in the medium and long-time horizons (Sagaert, Aghezzaf, Kourentzes & Desmet, 2018b, 2018a; Sharma, Singh & Singh, 2012). According to Verstraete et al. (2020), the impact of macroeconomic variables could be incorporated with two methods including manually adjusting the statistical forecast and expert forecast where both methods are expensive and biased.

Macroeconomic variables have been used in several studies of tactical sales forecasting using LASSO regression along with ARIMA models and non-linear ML methods attaining significant improvement in accuracy and it was found that traditional forecasting techniques such as regression illustrate poor performance over ML and shrinkage methods (Ludwig, Feuerriegel & Neumann, 2015; Sagaert et al., 2018b, 2018a). Moreover, the application of macroeconomic data for operational purposes is found challenging since data is more often published (Sagaert et al., 2018a). Hence, ML methods capable of incorporating several variables are more focused on sales prediction using various input variables. Among these predictive sales models built on ML, Adebayo (2018) has designed a Multilayer Feed Forward Neural Network (FFNN) along with a backpropagation algorithm comprised of 10 inputs and 10 nodes at the hidden layer to predict the sales of beer products and Carbonneau et al. (2008) has demonstrated the application of a NN model built with three layers feed-forward error back-propagation comprised of 5 inputs and using Hyperbolic tangent function as transfer function (Vhatkar & Dias, 2016).

Furthermore, researchers have evaluated the developed forecasting techniques integrated with multiple variables where Wang et al. (2019) has concluded that SVM is the best forecasting method for perishable products while LSTM is the best for non-perishable products considering evaluation index of overall performance while Shahrabi et al. (2009) has stated that ANN presents more persistent results while SVM performs better than the traditional forecasting techniques based on a comparison of forecasting results. Suzuki (2012) demonstrated ANN especially capable of identifying the most salient variables low weight for redundant and noise variables at training even at the presence of numerous variables as inputs, performs better than traditional and ML forecasting methods. Thus, the architecture of ANN can be exercised with any combination of fine predicting input variables with arbitrary flexibility, and it can be successfully trained.

Nonetheless, Guidolin et al., (2019); Huang et al., (2014); Ma et al., (2016) have assessed the effect of economic variables such as price and promotion focusing on the accuracy improvement of demand forecasts through various forecasting techniques. They concluded that the addition of economic variables adds value to the forecasting method by increasing its accuracy. No major study in the literature has been conducted using ANN models which has a significant impact on accuracy improvements with the addition of macroeconomic variables apart from price and promotion to assess its impact on the accuracy of demand forecasts.

Thus, this study is focused on building up an ANN model integrated with Price, Promotion and Macroeconomic variables to evaluate the accuracy of the model relative to the additional variables.

3. METHODOLOGY

The research aims to evaluate the effect of a macroeconomic variable in an ANN forecasting model. The feedforward error backpropagation method has proven to use with multiple variables which are chosen as the ANN structure. To assess the accuracy difference of the ANN model, Average Real Mean Absolute Error (*AvgRelMAE*) is preferred while Simple Exponential Smoothing is used as the benchmark forecasting model. The variables proposed for the process are Price and Promotion, and Federal Funds Rate (FFR) as a macroeconomic variable for the ANN model.

The basic structure of the feed-forward error backpropagation model consists of an input layer, hidden layer and output layer where the structure is mainly varied with the number of hidden layers and neurons in the model apart from the parameters such as activation function, batch size, loss value, number of epochs and dense layer value. By varying these parameters, the model structure can be modified in a way as to change forecast accuracy. Thus, it is important to define these parameters appropriately to create the best model (Goodfellow, Bengio & Courville, 2016).

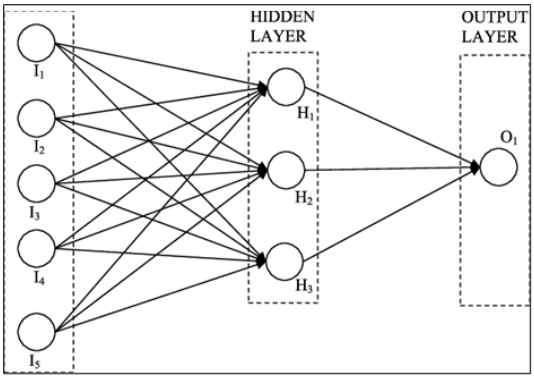


Fig. 1. Basic Structure of a NN (Srivastav, Sudheer & Chaubey, 2007)

The Initial ANN model (Model 1) proposed in the methodology is built only with price and promotion as variables. The second model (Model 2) is built by adding FFR value to the price and promotion variables to assess the accuracy improvement due to the addition of FFR values. The initial structure for both models contain 3 hidden layers. The input layer comprises neurons equal to the number of input variables and the output layer only with one neuron since forecast value is the only output as illustrated in Fig.1 while other parameters are defined accordingly with the data set as described below.

The hyper-parameters: number of neurons in the hidden layers, epochs, batch size and dense layer are defined according to the data with hyperparameter tuning either by manual adjusting by considering combinations defined by the modeller. But the identification of the best parameter values cannot be assured with manual adjustment since the manually evaluated combinations are less. Finding optimal values for the parameters can be automated with programmed functions to calculate the errors for all combinations. The common practice of defining some probable ranges for all the parameters and processing the automatic function within that range is exercised in this study. Thus, a considerable number of combinations is to be tested to find the best hyperparameter values.

In evaluating the results, the accuracy measuring method plays a vital role with the presence of many methods with several drawbacks in each error measuring method. *AvgRelMAE* which uses a benchmark model to compare the selected forecasting method is being suggested as the most suitable method by Davydenko & Fildes, (2016) with practical recommendations. It is chosen to compare the accuracy changes and the calculations will be done based on the following equations.

For each time series i in 1…m:

$$r_i = \frac{MAE_i^A}{MAE_i^{B'}} \tag{1}$$

where: *MAE* – Mean Absolute Error,
γ_i – relative MAE,
A – proposed forecasting model,
B – benchmark forecasting model.

$$r_i \, l_i = n_i \ln r_i \tag{2}$$

$$AvgRelMAE = exp^{\left(\frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} l_i\right)} \tag{3}$$

The *AvgRelMAE* values of Model 1 and Model 2 are compared with the *AvgRelMAE* value of the benchmark model. *AvgRelMAE* of the benchmark model is 1. If the *AvgRelMAE* of any other model is higher than 1, it concludes that the accuracy of the proposed method has been reduced over the benchmark model.

4. DATA ANALYSIS AND RESULTS

Data was extracted from James M. Kilts Center; University of Chicago Booth School of Business which was selected from the freely available data sources. The set of data has been collected from the company, Dominick's Finer Foods (DFF) inclusive of more than 25 categories and 100 store chains. 5 Data sets were chosen from 5 categories based on the number of weeks available. Tab.1 consists of the details of the selected Universal Product Codes (UPCs).

Tab. 1. Selected UPCs

Category	UPC Number	Name	Weeks	Tag
Frozen Entrees	1380010068	STFRS SWEDISH MTBALL 11 OZ	396	UPC1
Refrigerated Juices	3828154001	HH ORANGE JUICE 64 OZ	396	UPC2
Front-end-candies	4000000102	SNICKERS 1 CT	396	UPC3
Frozen Juices	3828190029	HH ORANGE JUICE CONC 12 OZ	396	UPC4
Cheeses	2100061223	KR PHILA CREAM CHEESE 8 OZ	392	UPC5

The raw data of daily sales of a product in each shop was aggregated to the units sold within a week in all the stores. Due to the promotions and other possible reasons such as inflation, the price of a product has a range in the time horizon. Therefore, the unit price has been calculated by converting the price distribution for the promotions into a standard normal distribution. Also, there were 3 types of promotion, and a promotional index has been introduced to evaluate the power of promotions in a particular week. The ratio, the quantity sold under any promotional type divided by the total units sold in a has been calculated as the promotional index to reflect the power of the promotion across all stores. It reflects the percentage of units sold under any promotion. After calculating the sold quantity, unit price, promotional index, the macroeconomic value is added to the model. Federal Funds Rate (FFR) in the USA has been chosen as the macroeconomic variable and has been merged with the relevant week of the data set. The sample of a data set after refining is as shown in Tab. 2.

Tab. 2. Sample Data View

Quantity	Price	Promotion	FFR
994	0.505254	0	8.25
1030	0.505254	0	8.27
4838	-1.26969	1	8.28
871	0.505254	0	8.27
936	0.505254	0	8.27
836	0.549627	0	8.26

The 0.8 to 0.2 split has been used for the training and testing data sets where 317 data points were used to predict 80 weeks sales. The training data set has been rescaled taking the mean as the centre and standard deviation as the scale (Z score method). The mean of the training data set and standard deviation of the training data is used in the test data for normalization since the test data is only used for validation purposes.

The ANN models have been created using the R software. The basic model for Model 1 and 2 is made with one input layer, one output layer and three hidden layers. Input shape is the number of inputs varying from 3 to 4 inputs whether the model is using the FFR value or not. The models have been created using the R project for statistical computing. Fig. 2 shows a basic plot of a neural net of a model 2 which includes all three inputs. 3 hidden layers have been used in this model.

Rectified linear unit activation function has been used as the activation function for the input layers and hidden layers and MAE has been used as the loss function. The initial model had 50 epochs and the batch size was 4. After creating the initial model, the hyperparameter tuning is done by changing each variable. Below are the hyperparameters which are tuned to find the best accuracy.

1. The number of neurons in the 1st hidden layer.
2. The number of neurons in the 2nd hidden layer.
3. The number of neurons in the 3rd hidden layer.
4. The dropout value of the 1st hidden layer.
5. The dropout value of the 2nd hidden layer.

6. The dropout value of the 3rd hidden layer.
7. The number of batches.
8. The number of epochs.

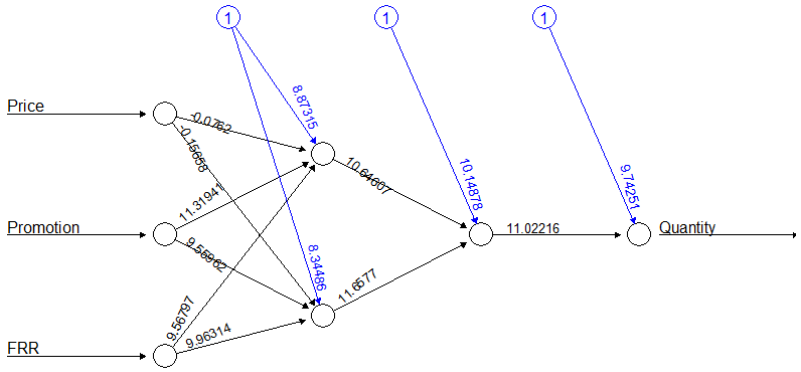


Fig. 2. Neural net plot of basic Model 2 structure

The hyperparameters were selected based on the common practice of testing in each parameter in relevant ranges. To get a value of a one parameter, it was selected as a variable while others were taken as non-variables where the variable was tested for a range comparing the error. Consequently, the values for each parameter were chosen based on the same method. The hyperparameters of the UPC 1 is shown in the table.

Tab. 3. Hyperparameters of UPC1 Models

Parameter	1st	2nd	3rd	4th	5th	6th	7th	8th
UPC1 Model 1	50	60	70	0.3	0.2	0.1	2	40
UPC1 Model 2	80	40	70	0.3	0.2	0.1	2	40

After calculating optimal hyperparameter values, the forecasted results and the error has been calculated separately. Since there are 5 data sets, 10 ANN models were created for both Model 1 and 2. The final results are focused on two accuracy comparisons that need to be assessed using the error measuring method, as mentioned below:

1. Benchmark model (SES) vs Model 1 (Price & promotion).
2. Benchmark model (SES) vs Model 2 (Price & promotion & FFR).

Figure 3 below is a plot of the forecast values of Model 1, Model 2 and simple exponential smoothing with the actual value. Based on the plot, it can be clearly seen that Model 2 has over forecasted than the Model 1 where Model 1 is the closest to the actual value for UPC1. Since there are 5 UPCs considered, as explained in the methodology, *AvgRelMAE* has been used to mathematically compare the models.

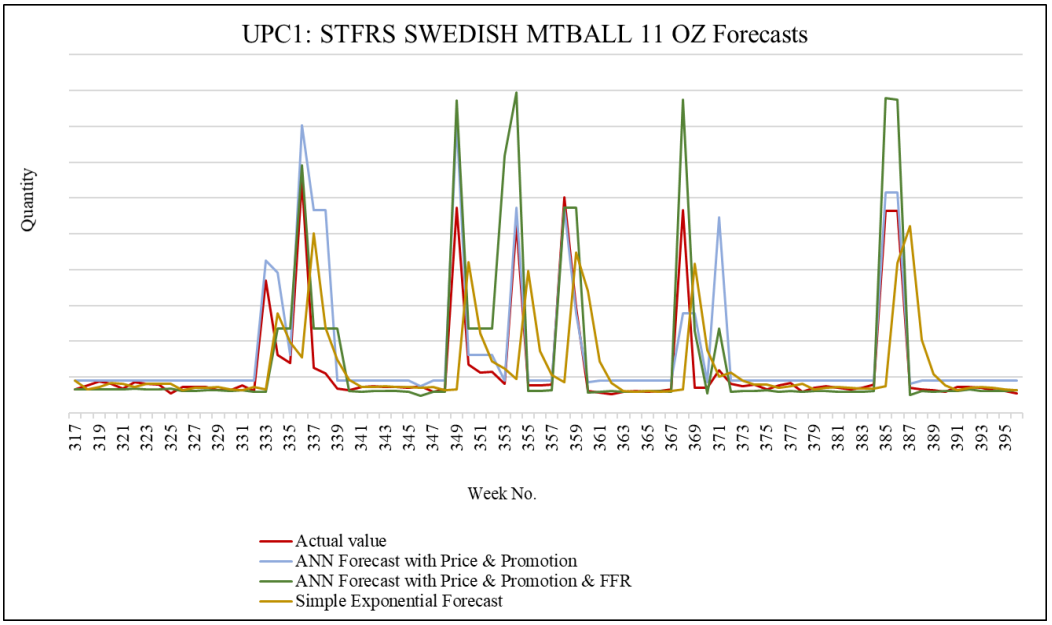


Fig. 3. Plot of forecasts for UPC1

Since there are 5 UPCs; i = 1,2,3,4,5 for all i; n_i = 80. The r_i and l_i have been calculated for all i using Equation 1, Equation 2 and Equation 3 as shown in a previous paragraph. The results of the comparisons are presented in Tab.4.

Tab. 4. Final *AvgRelMAE* Values

	Simple Exponential Forecast	**ANN Forecast with Price & Promotion**	**ANN Forecast with Price & Promotion & FFR**
AvgRelMAE	1.00	0.68	–
	1.00	–	0.71
	–	1.00	1.05

According to Davydenko & Fildes (2016), the values are compared against 1, which is the *AvgRelMAE* of the benchmark model. If *AvgRelMAE* is lower than 1, the accuracy is improved. According to Tab.4, it is evident that the ANN model with price & promotion has a higher forecast accuracy over simple exponential smoothing. Although the second model has a lesser value than 1, it is still higher than the initial model with only price and promotion. Therefore, the *AvgRelMAE* can be interpreted that adding the Federal Funds Rate to the initial model with Price and Promotion has not increased the forecast accuracy, rather it has decreased the forecast accuracy.

5. CONCLUSIONS

The possible variables which could impact the accuracy were identified and the performance of ANN models was confirmed through a comprehensive literature study, thus, the ANN-based models with selected multiple variables were developed. A feed-forward back-propagation type ANN model has been identified as the best ANN method for regression type models. The models were built using data extracted from the James M. Kilts Center, University of Chicago Booth School of Business, which was collected from Dominick's Finer Foods, in California, USA. The data consisted of weekly sales of many UPCs of various FMCG product categories, economic information on price and the promotion and some other information. The price and promotional data were filtered and used as two variables of the model. In addition to this, the FFR, which is a macroeconomic indicator of interest rate, has also been used as a variable in creating the models. Thus, price, promotion and FFR were used as multiple variables in building the ANN models. 5 data sets of 5 different UPCs were used in this research which was filtered and selected based on the availability of the data and sales volume. Two models were developed for one data set including an ANN model with price and promotional data and another ANN model with price, promotion and FFR data as variables. It resulted in 10 ANN. The structure of the ANN model was obtained through hyperparameter tuning. The number of neurons in hidden layers, epochs, batch size, dense were defined using hyperparameter tuning for each model. The combination of these parameters which resulted in the least error was taken to measure the accuracy in the following step.

This study covers five simple exponential smoothing models that were created using historical data using Microsoft Excel to be used as baseline models. As the initial accuracy measuring method, MAE was taken for all the models. Assessing the accuracy was done using the Average Relative Mean Absolute Error (*AvgRelMAE*). This method combines all the UPC error rates and gives an overall accuracy comparison of one method over another method. The comparison of the benchmark model, simple exponential smoothing and ANN model with price and promotion concludes that ANN model forecasts are much accurate. Also, it is concluded that the accuracy of the ANN model with price promotion and FFR value is higher than the simple exponential smoothing. Although there is an accuracy improvement of the ANN model with Price, Promotion and FFR value over the benchmark model, the study finds that the accuracy has decreased when adding FFR value to the ANN model.

This study has mainly focused on the interest rate as a macroeconomic variable and can be different variables such as GDP, unemployment rate and many other macroeconomic variables which could be considered as variables in this method of modelling. Also, there are many machine learning methods other than the Feedforward Neural Network model to incorporate any number of variables. In addition to that, assessing the accuracy improvement by the power of the sales force, advertising power and other economic variables integrated with ML models could be researched in further studies. The models were only developed for some products which can be further improved to assess the possibility of creating demand forecasts for the products in other domains using a similar methodology.

Funding

This research was funded by the Senate Research Committee Grant ID SRC/LT/2021/22, University of Moratuwa, Sri Lanka.

REFERENCES

Abolghasemi, M., Eshragh, A., Hurley, J., & Fahimnia, B. (2020). Demand Forecasting in the Presence of Systematic Events: Cases in Capturing Sales Promotions. *International Journal of Production Economics*, *230*, 107892. https://doi.org/10.1016/j.ijpe.2020.107892

Adebayo, A. (2018). Predictive Sales Model using Multi-layer Neural Network with Backpropagation Algorithm. *International Journal of Engineering Technology, Management and Applied Sciences*, *6*(4), 30–40.

Ali, Ö. G., Sayin, S., van Woensel, T., & Fransoo, J. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*(10), 12340–12348. https://doi.org/10.1016/j.eswa.2009.04.052

Balachandra, K., Perera, H. N., & Thibbotuwawa, A. (2020). Human Factor in Forecasting and Behavioral Inventory Decisions: A System Dynamics Perspective. In *International Conference on Dynamics in Logistics* (pp. 516–526). Springer, Cham. https://doi.org/10.1007/978-3-030-44783-0_48

Barker, J. (2020). Machine learning in M4 : What makes a good unstructured model? *International Journal of Forecasting*, *36*(1), 150–155. https://doi.org/10.1016/j.ijforecast.2019.06.001

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research*, *184*(3), 1140–1154. https://doi.org/10.1016/j.ejor.2006.12.004

Davydenko, A., & Fildes, R. (2016). Forecast Error Measures : Critical Review and Practical Recommendations. In *Business Forecasting: Practical Problems and Solutions*. John Wiley & Sons Inc. https://doi.org/10.13140/RG.2.1.4539.5281

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*, *in press*. https://doi.org/10.1016/j.ijforecast.2019.06.004

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Guidolin, M., Guseo, R., & Mortarino, C. (2019). Regular and promotional sales in new product life cycles: Competition and forecasting. *Computers and Industrial Engineering*, *130*, 250–257. https://doi.org/10.1016/j.cie.2019.02.026

Harris, N. L., Nadler, L. M., & Bhan, A. K. (1984). Review of Nils Nilsson Principles of Artificial Intelligence. *The American Journal of Pathology*, *117*(2), 262-272. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/6437232%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1900435

Hewage, H. C., Perera, H. N., & De Baets, S. (2021). Forecast adjustments during post-promotional periods. *European Journal of Operational Research*, *in press*. https://doi.org/10.1016/j.ejor.2021.07.057

Huang, T., Fildes, R., & Soopramanien, D. (2014). The value of competitive information in forecasting FMCG retail product sales and the variable selection problem. *European Journal of Operational Research*, *237*(2), 738–748. https://doi.org/10.1016/j.ejor.2014.02.022

Huang, T., Fildes, R., & Soopramanien, D. (2019). Forecasting retailer product sales in the presence of structural change. *European Journal of Operational Research*, *279*(2), 459–470. https://doi.org/10.1016/j.ejor.2019.06.011

Ludwig, N., Feuerriegel, S., & Neumann, D. (2015). Putting Big Data analytics to work: Feature selection for forecasting electricity prices using the LASSO and random forests. *Journal of Decision Systems*, *24*(1), 19–36. https://doi.org/10.1080/12460125.2015.994290

Ma, S., Fildes, R., & Huang, T. (2016). Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research*, *249*(1), 245–257. https://doi.org/10.1016/j.ejor.2015.08.029

Matharage, S. T., Hewage, U., & Perera, H. N. (2020). Impact of Sharing Point of Sales Data and Inventory Information on Bullwhip Effect. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (pp. 857–861). IEEE. https://doi.org/10.1109/IEEM45057.2020.9309733

Ni, D., Xiao, Z., & Lim, M. K. (2019). A systematic review of the research trends of machine learning in supply chain management. *International Journal of Machine Learning and Cybernetics*, *11*, 1463–1482. https://doi.org/10.1007/s13042-019-01050-0

Oliva, R., & Watson, N. (2009). Managing functional biases in organizational forecasts: A case study of consensus forecasting in supply chain planning. *Production and Operations Management*, *18*(2), 138–151. https://doi.org/10.1111/j.1937-5956.2009.01003.x

Parker, S. (2014). Principles and Practice. *IFLA Journal*, *32*(3), 179-180. https://doi.org/10.1177/0340035206070163

Perera, H. N., & Sudusinghe, J. I. (2017). Longitudinal analysis of supply chain transformation project management. *2017 Moratuwa Engineering Research Conference (MERCon)* (pp. 153–158). IEEE. https://doi.org/10.1109/MERCon.2017.7980473

Perera, H. N., Thibbotuwawa, A. I., Rajasooriyar, C., & Sugathadasa, P. R. S. (2016). Managing Supply Chain Transformation Projects in the Manufacturing Sector: Case-based Learning from Sri Lanka. In *Conference on Research for Transportand Logistics Industry 2016* (pp. 143–145). R4TLI-D13.

Sagaert, Y. R., Aghezzaf, E. H., Kourentzes, N., & Desmet, B. (2018a). Tactical sales forecasting using a very large set of macroeconomic indicators. *European Journal of Operational Research*, *264*(2), 558–569. https://doi.org/10.1016/j.ejor.2017.06.054

Sagaert, Y. R., Aghezzaf, E. H., Kourentzes, N., & Desmet, B. (2018b). Temporal big data for tactical sales forecasting in the tire industry. *Interfaces*, *48*(2), 121-129. https://doi.org/10.1287/inte.2017.0901

Shahrabi, J., Mousavi, S. S., & Heydar, M. (2009). Supply chain demand forecasting: A comparison of machine learning techniques and traditional methods. *Journal of Applied Sciences*, *9*(3), 521–527. https://doi.org/10.3923/jas.2009.521.527

Sharma, G. D., Singh, S., & Singh, G. S. (2012). Impact of Macroeconomic Variables on Economic Performance: An Empirical Study of India and Sri Lanka. *SSRN Electronic Journal*, 1-35. https://doi.org/10.2139/ssrn.1836542

Spiliotis, E., Makridakis, S., Semenoglou, A. A., & Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research*. https://doi.org/10.1007/s12351-020-00605-2

Srivastav, R., Sudheer, K., & Chaubey, I. (2007). A simplified approach to quantify predictive and parametric uncertainty in artificial neural network hydrologic models. *Water Resour*, *43*(10), W10407. https://doi.org/10.1029/2006WR005352

Ranil, P. T., Sugathadasa, S., Senadheera, S. W., & Thibbotuwawa, A. (2021). A Study of Supply Chain Risk Factors of the Large-Scale Apparel Manufacturing Companies–Sri Lanka. *Engineer*, *54*(03), 49–58. http://doi.org/10.4038/engineer.v54i3.7459

Sugathadasa, R., Wakkumbura, D., Perera, H. N., & Thibbotuwawa, A. (2021). Analysis of Risk Factors for Temperature-Controlled Warehouses. *Operations and Supply Chain Management: An International Journal*, *14*(3), 320–337. http://doi.org/10.31387/oscm0460305

Suzuki, K. (2012). *Artificial Neural Networks. Methodological Advances and Biomedical Applications*. IntechOpen.

Tangjitprom, N. (2012). The Review of Macroeconomic Factors and Stock Returns. *International Business Research*, *5*(8), 107–115. https://doi.org/10.5539/ibr.v5n8p107

Verstraete, G., Aghezzaf, E. H., & Desmet, B. (2020). A leading macroeconomic indicators' based framework to automatically generate tactical sales forecasts. *Computers and Industrial Engineering*, *139*, 106169. https://doi.org/10.1016/j.cie.2019.106169

Vhatkar, S., & Dias, J. (2016). Oral-Care Goods Sales Forecasting Using Artificial Neural Network Model. *Procedia Computer Science*, *79*, 238–243. https://doi.org/10.1016/j.procs.2016.03.031

Wang, P.-H., Lin, G.-H., & Wang, Y.-C. (2019). Applied Sciences Application of Neural Networks to Explore Manufacturing Sales Prediction. *Applied Sciences*, *9*(23), 5107. https://doi.org/10.3390/app9235107

Yang, D., Goh, G. S. W., Xu, C., Zhang, A. N., & Akcan, O. (2015). Forecast UPC-level FMCG demand, Part I: Exploratory analysis and visualization. *Proceedings – 2015 IEEE International Conference on Big Data* (pp. 2106–2112). IEEE. https://doi.org/10.1109/BigData.2015.7363993

Applied Computer Science, vol.17, no. 4, pp. 45–54
doi: 10.23743/acs-2021-28

Submitted: 2021-07-11 | Revised: 2021-08-14 | Accepted: 2021-09-20

Keywords: stuttering, fillers disfluency, automatic recognition, fillers detection

Waldemar SUSZYŃSKI [0000-0003-2990-2078]*,
Małgorzata CHARYTANOWICZ [0000-0002-1956-3941]*,
Wojciech ROSA [0000-0002-7051-6008]**, *Leopold KOCZAN* [0000-0002-7775-1836]**,
Rafał STĘGIERSKI [0000-0001-7225-3275I]*

DETECTION OF FILLERS IN THE SPEECH BY PEOPLE WHO STUTTER

Abstract

Stuttering is a speech impediment that is a very complex disorder. It is difficult to diagnose and treat, and is of unknown initiation, despite the large number of studies in this field. Stuttering can take many forms and varies from person to person, and it can change under the influence of external factors. Diagnosing and treating speech disorders such as stuttering requires from a speech therapist, not only good professional preparation, but also experience gained through research and practice in the field. The use of acoustic methods in combination with elements of artificial intelligence makes it possible to objectively assess the disorder, as well as to control the effects of treatment. The main aim of the study was to present an algorithm for automatic recognition of fillers disfluency in the statements of people who stutter. This is done on the basis of their parameterized features in the amplitude-frequency space. The work provides as well, exemplary results demonstrating their possibility and effectiveness. In order to verify and optimize the procedures, the statements of seven stutterers with duration of 2 to 4 minutes were selected. Over 70% efficiency and predictability of automatic detection of these disfluencies was achieved. The use of an automatic method in conjunction with therapy for a stuttering person can give us the opportunity to objectively assess the disorder, as well as to evaluate the progress of therapy.

1. INTRODUCTION

The recognition of speech pathology on the basis of the acoustic analysis of the utterance enables simple, non-invasive diagnostics. Stuttering is a speech impediment that is very complex, difficult to diagnose and treat, and also not fully understood, despite the large number of studies in this field. It can take many forms and can vary from person to person. Moreover, it can ease off or intensify under the influence of external factors (Bloodstein 1995; Stromsta1993, Wingate 2012).

* Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, Poland, w.suszynski@pollub.pl, m.charytanowicz@pollub.pl, rafal.stegierski@gmail.com
** Lublin University of Technology, Faculty of Technology Fundamentals, Poland, w.rosa@pollub.pl, l.koczan@pollub.pl

Measurements of disfluent episodes in the speech of people who stutter are very important in diagnosing, monitoring the course and assessing the final effects of therapy. Currently, they are auditioned, which is associated with a considerable effort on the part of the speech therapist. Due to the subjectivity of such measurements and sometimes low agreement in the assessments of the audience, it is difficult to compare different therapeutic techniques. It would be desirable, therefore, to develop an objective method of detecting and measuring the duration of individual types of disfluencies, based on the acoustic characteristics of the speech signal (Kuniszyk-Jóźkowiak et al., 2003, 2004).

In this article, we focus on one type of stuttering – fillers. They are among the mildest symptoms of stuttering. In Polish, the vowel "y" is most often used as a filler. It is often an approach taken by people who stutter as a way to begin the utterance of a difficult word. Fillers are common in fluent speech as well, and if rare enough, they do not affect the overall judgment of the statement. In the case of people who stutter, this relatively mild form of disfluency should be controlled and gradually eliminated, as it often hides therapeutically improper treatment that does not eliminate the psychological basis of this extremely complicated disorder.

Automatic determination of the fluency disturbance level is very significant for diagnosing, forecasting and therapy, and the detection and duration measurements of stuttering episodes are of great importance in a logopaedist's work. However, there are not many studies aiming at automation of speech assessment of people who stutter. Still, such studies were carried out by Howell and colleagues (Howell A Sackin,1995; Howell et al., 1997; Czyżewski, Kaczmarek & Kostek, 2003; Kuniszyk-Jóźkowiak, Smołka & Suszyński, 2001). In other studies ware used fuzzy logic (Suszynski et al., 2003a, 2003b), correlation function (Suszynski et al., 2005), Hidden Markov Models (Wiśniewski et al., 2010; Wiśniewski & Kuniszyk-Jóźkowiak, 2015), Kohonenn Neural Network (Smołka et al., 2003) or Hierarchical AAN system (Świetlicka, Kuniszyk-Jóźkowiak & Smołka, 2013). It was also used label sequences to detect stuttering events in reading speech (Alharbia et al., 2020).

The aim of our study was to develop an algorithm for automatic recognition of fillers in the statements of people who stutter – doing so on the basis of their parameterized features in the amplitude-frequency space.

2. PREPARATION OF MATERIAL FOR RESEARCH

The acoustic classification and identification of stuttering was carried out by observing the spectral waveforms and parameters obtained from the developed computer procedures. Based on the results of the acoustic classification of disfluency, separate procedures have been developed for the recognition and classification of individual groups of these episodes. The acoustic features of particular types of disfluency and the limits of their variability were determined on the basis of a set of disfluent statements of stutterers and a comparison of these with their fluent counterparts.

The simplest and most frequently used graphic image of speech signals is an oscillograph record. The program for acquiring and processing recordings on the oscillograph enables the initial visualization of speech, marking or deleting specific fragments, adjusting the time scale and amplitude, etc. In the diagnosis of many speech disorders, as well as in work with people with hearing problems, amplitude recording alone is not sufficient. Full information

about speech signals is given by three-dimensional frequency characteristics that take in the variables of time, amplitude and frequency. This analysis shows the changes taking place in speech with distorted articulation. However, it requires some experience on the part of the analysing person to be accomplished in this field.

The research used the digital speech signals of people who stutter. The data were analyzed by FFT with a Hamming window at 20 ms time intervals using $N = 21$ one-third octave filters in the frequency range of 100-10000 Hz. Additionally, an A filter was used and a logarithmic amplitude scale was applied. This type of analysis is a certain approximation of the characteristics of sound processing by the human auditory system. This approach to analysis allowed the development of automatic methods similar to human analysis (Moore & Glasberg, 1983; Moore, Peters & Glasberg,1990). In addition, for the automation of the process, the average sound levels and the band in which the maximum sound level was located were determined.

A general block diagram of a set of computer procedures is shown in Fig. 1. We can distinguish here outputs for preliminary analyzes and an automatic detection block. In the test block of the program, the time courses of the average level and the location of the maximum spectrum were determined and visualized. In the automatic detection block, various types of disfluency were detected and classified. Herein, four main types of non-fluent episodes were distinguished: prolongation, stops, repetitions and fillers (Fig. 1).

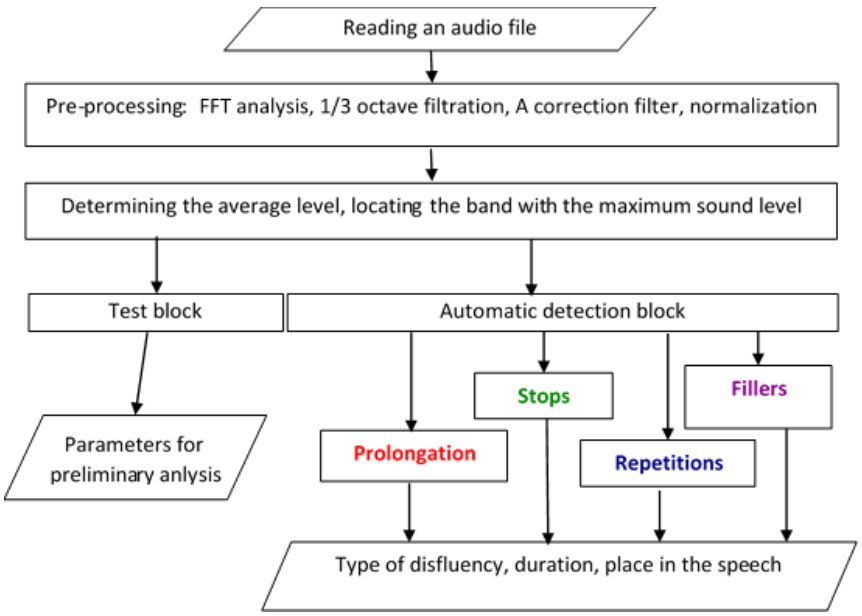


Fig. 1. General block diagram of the program for the analysis and automatic recognition of speech disfluency

3. ALGORITHM AND OPERATION OF THE FILLERS DETECTION PROGRAM

The fillers detection program includes the following procedures:

1. Calculation and visualization of the 1/3 octave spectrum.
2. Calculation and visualization of the average sound level.
3. Arbitrary stretching and narrowing of these waveforms and reproduction of the sound, the spectrum and average level of which are illustrated.
4. Readout of the cursor position on the average level waveform, which is set by the user of the program at the beginning of the filler considered being characteristic of the given person. Setting and reading the final position of the cursor, which practically comes down to choosing the width of the time window T.
5. Calculation of the correlation function according to formula (1), (2).

The principle of fillers detection was the correlation with the pattern marked by the examiner. Based on the conducted research, it can be concluded that in the majority of people who stutter, if there are any interferences, the outcome of the applied research refers to the same that is characteristic for a given person's sound (the "y" is most often inserted in the Polish language).

Calculation of the correlation function according to the following formulae:

$$R(t,T) = \frac{\sum_{i=1}^{N}\sum_{l=0}^{T-1}[x_i(t+l)-\mu(t)][x_i(t_w+l)-\mu(t_w)]}{\sqrt{\left(\sum_{i=1}^{N}\sum_{l=0}^{T-1}[x_i(t+l)-\mu(t)]^2\right)\left(\sum_{i=1}^{N}\sum_{l=0}^{T-1}[x_i(t_w+l)-\mu(t_w)]^2\right)}} \quad (1)$$

$$\mu(t) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t}^{t+T-1} x_i(t) \quad (2)$$

where: $x_i(t), x_i(t_w)$ – corrected sound levels in the i-th band for the time samples t and pattern t_w,
$\mu(t)$, $\mu(t_w)$ – average of all values in N bands in the window T,
t – current number of the time sample,
T – window width (number of samples in the time window),
l – number of the sample in the window.

Figure 2 shows an example of the obtained data set by pre-processing sound samples. The table contains the corrected and normalized sound levels in the bands 1/3-octave at consecutive moments of time. The columns represent successive moments of time (marked with t1, t2, ..., t29), lines – successive 1/3-octave bands (1–21). Expert selected patterns with a window width of two (t11–t12) are marked in red, while the current time window, shifting from the beginning to the end of the file is in green. Left arrow indicates the computation of the correlation for t = t1, right arrow for t = t17. The correlation coefficient described by formula 1 for a given pattern (width T) is only a function of time.

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14	t15	t16	t17	t18	t19	t20	t21	t22	t23	t24	t25	t26	t27	t28	t29
1	46	20	48	47	44	49	27	25	19	25	17	26	7	47	9	40	19	32	45	30	0	37	40	32	25	48	29	42	29
2	3	11	25	7	3	6	18	28	29	22	12	30	23	37	22	20	2	48	9	2	29	17	2	47	19	47	21	48	7
3	29	31	48	49	47	45	5	6	37	3	37	21	4	42	10	31	16	24	3	33	43	38	27	46	19	25	3	7	13
4	48	31	13	38	38	10	39	36	25	2	26	41	23	40	41	43	48	25	9	27	44	23	15	5	15	18	7	1	4
5	46	3	14	23	19	24	9	5	47	7	32	1	43	16	20	37	13	29	7	26	34	49	47	39	49	26	37	47	7
6	49	28	11	26	16	0	19	13	38	36	41	28	40	29	6	34	17	8	46	26	34	37	8	46	37	24	33	47	34
7	17	33	21	0	29	1	0	37	2	46	22	49	46	26	29	8	44	24	29	46	28	18	27	9	22	40	2	6	28
8	25	47	36	48	23	23	48	1	41	46	20	3	14	13	10	0	12	27	47	36	0	0	37	4	45	6	38	35	33
9	48	0	11	27	20	12	39	20	40	14	36	13	4	13	12	34	38	10	21	9	20	20	10	21	1	18	43	42	25
10	40	16	44	12	24	25	29	20	37	22	24	18	34	42	46	1	37	35	30	32	2	15	7	20	43	42	24	3	0
11	29	38	34	25	17	20	16	11	40	1	25	17	16	6	28	2	5	10	38	29	43	38	38	30	37	26	5	12	10
12	21	5	42	18	3	8	18	40	11	22	15	3	47	6	48	44	31	39	46	7	26	38	17	44	15	37	21	33	11
13	13	41	35	20	31	42	13	22	19	33	41	39	44	45	27	40	4	0	14	28	39	49	22	45	20	14	31	16	32
14	21	41	19	46	3	42	28	30	33	35	20	20	38	17	34	35	28	8	7	1	36	45	19	0	7	27	41	8	24
15	49	7	19	7	11	29	28	9	49	39	1	45	2	20	44	2	6	34	10	34	1	25	4	42	43	2	39	15	0
16	32	24	45	48	30	49	36	38	37	47	48	4	38	0	22	49	25	27	1	35	29	25	25	5	6	44	16	13	20
17	39	28	23	19	48	2	20	24	7	18	9	23	3	20	23	2	18	25	18	21	5	8	48	6	3	39	0	21	40
18	33	14	7	41	6	3	3	45	15	21	44	25	8	22	15	5	19	27	9	44	9	36	40	43	12	49	13	31	30
19	0	14	39	0	31	23	0	5	29	41	33	37	21	19	47	5	5	25	45	4	15	33	18	40	48	36	22	27	29
20	16	13	10	28	0	45	44	32	10	37	1	4	32	25	27	44	43	28	43	41	33	18	7	28	15	7	22	3	12
21	7	39	29	16	31	30	7	22	23	28	32	34	28	5	11	36	42	13	47	15	39	49	26	22	17	44	15	44	38

Fig 2. Schematic diagram of the correlation procedure
(vertical – consecutive spectra, horizontal – consecutive time moments)

In Figure 3, we can see a screenshot used to automatically search for fillers in a stutterer's statements.

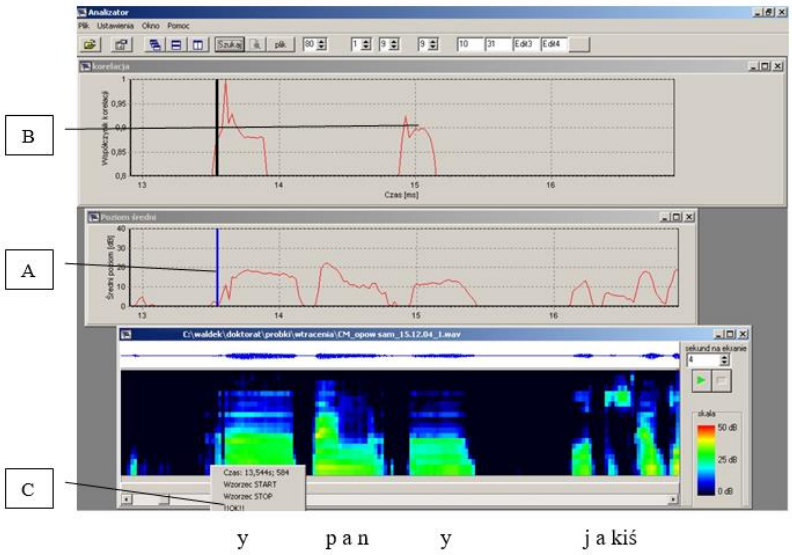


Fig. 3. Screen from the program for automatic fillers detection
(statement in Polish "pan jakiś" with fillers "y")

The top window – the value of the correlation coefficient; middle window – medium sound level; lower window – oscillogram and spectrogram. A – setting the beginning of the pattern, B – correctly identified repeated fills, C – time (from the beginning of the sound file) indicated by the cursor position.

Due to the variety of fillers present, initial expert input was necessary. The individual had to indicate in the audio file one characteristic occurrence of a given disorder. The program then automatically indicated other instances of this disfluency. The examiner had to mark the beginning of the filler and determine its length (Figure 3). A spectrogram (lower window) was presented in the program window facilitating speech analysis. There was also the possibility to play a fragment of a speech sample.

In Part B of this figure, the expert observed the operation of the algorithm – the chart presents the value of the correlation coefficient (at the beginning of the chart it can be observed – the filler is detected and the correlation coefficient reaches the maximum value of one). The second peak (approx. 15 s) indicates the next filler detected (the correlation coefficient is approx. 0.8). Measurements were made for different sizes of windows. The obtained data were saved and on their basis 3D charts were prepared (Figures 4–6).

The program was created in the Delphi environment and written in the Pascal language. All procedures are implemented directly in the code.

4. VERIFICATION

As part of the research, the sensitivity and predictability of the method were determined. The dependence on the border coefficient of correlation and the width of the time window on the above parameters was also assessed. Setting the pattern start also plays an important role in detecting disfluency. Through many experiments, it was found that the optimal positioning of the cursor at the beginning of the disfluency chosen as the reference is (formula 3 and formula 4).

$$sensitivity = \frac{\sum correctly\ detected\ fillers}{\sum correctly\ detected\ fillers + \sum\ undetected\ fillers} \tag{3}$$

$$predictability = \frac{\sum correctly\ detected\ fillers}{\sum correctly\ detected\ fillers + \sum\ false\ fillers} \tag{4}$$

In order to verify and optimize the proposed procedures, the statements of seven stutterers with total of 170 fillers were selected. Each file contained one or more fillers surrounded by fluently spoken words. The examiner was responsible for selecting the model disfluency. The aim of these studies was to select parameters so that both the sensitivity and predictability were as high as possible and the correlation coefficient was at a level clearly indicating the similarity of the fillers found. Contour charts were built on the basis of the obtained data in order to accurately trace and select the optimal parameters. The optimal parameters of the tested parameters were those for which sensitivity and predictability exceed at least 70%. In the drawings, they were located in the ranges covered by red lines and marked in yellow.

Figures 4–7 show exemplary contour graphs of sensitivity and predictability depending on the boundary value of the correlation coefficient and the width of the time window for different people and different fillers. Each of the figures at the top presents also an example of the settings of the start and end of the filler pattern on the average sound level, along with an example of the length of the time window for which the assumed predictability and sensitivity results were obtained.

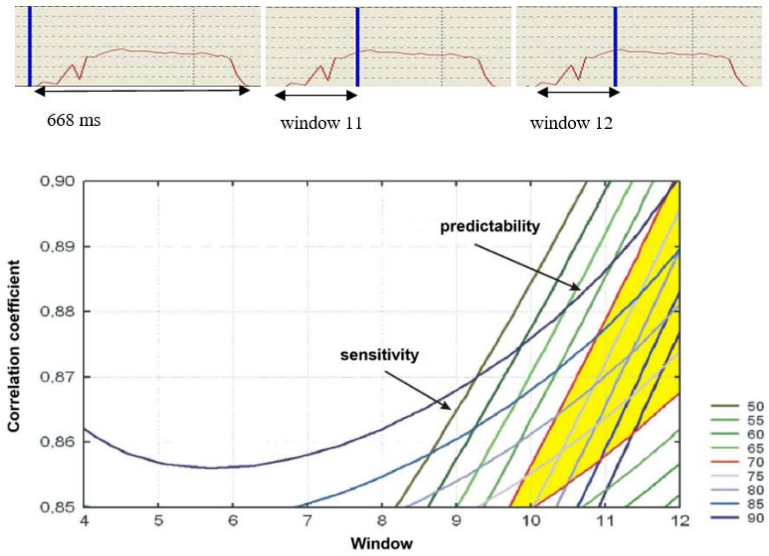


Fig. 4. Contour plots of the dependence of sensitivity and predictability on the limit value of the correlation coefficient and the width of the time window – case 1: correct identification for long windows

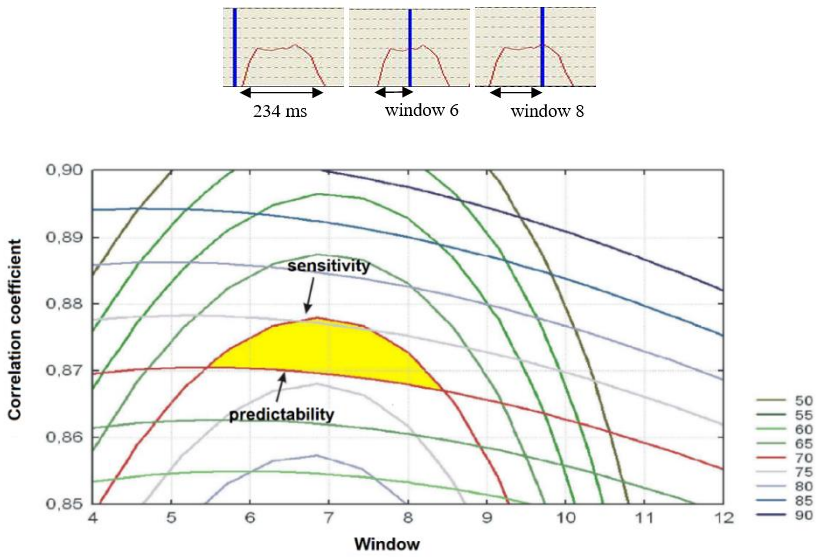


Fig. 5. Contour plots of the dependence of sensitivity and predictability on the limit value of the correlation coefficient and the width of the time window – case 2: correct identification for short windows

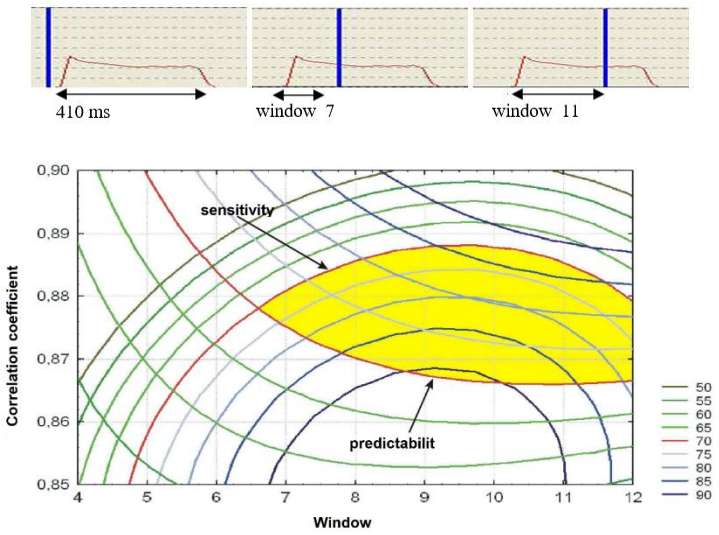


Fig. 6. Contour plots of the dependence of sensitivity and predictability on the limit value of the correlation coefficient and the width of the time window – case 3: correct identification for average windows

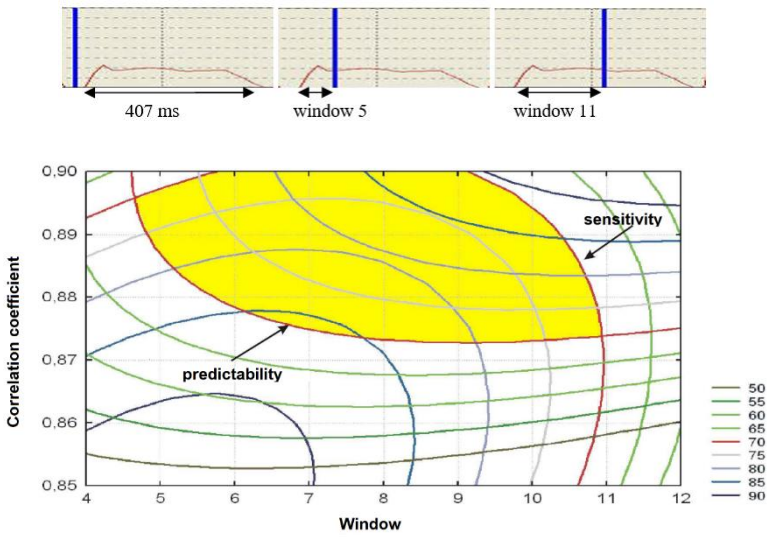


Fig. 7. Contour plots of the dependence of sensitivity and predictability on the limit value of the correlation coefficient and the width of the time window – case 4: correct identification for average windows with a simultaneous large value of the correlation function

The speech disorder described at work is difficult to detect automatically and requires a preliminary precise determination of the disfluency pattern and the method of searching for subsequent episodes. The presented drawings show a fragment of the research leading to the results, as well as to illustrate various cases. Figure 4 presents the results for a long filler which is correctly detected only for long windows. Figure 5 – a short filler – positive results

were obtained for short windows. In Figure 6, an example of average length of an filler – a wide range of windows for positive identification. Figure 7 – medium-length with a very large correlation function.

The width of the time window is specified as times of 23 ms. The optimal choice of the time window width was assumed so that both parameters exceed 70%. The areas corresponding to the optimal selection of parameters are highlighted.

Data analysis allowed to establish that the limit that should be set for the correlation coefficient should be 0.87–0.88. As can be seen, the width of the time window (the final position of the cursor on the pattern image) does not need to be precisely defined. However, it should be inserted in the middle of the pattern (between 1/3 and 2/3 of its duration).

5. SUMMARY

The developed and described procedures for automatic recognition of this type of disfluency can be used in continuous speech and do not require initial segmentation. According to verification, they also do not introduce erroneous classifications of the disfluency type and do not require perfect noise-free audio recordings.

In order to verify and optimize the procedures, the statements of seven stutterers (four boys and three girls aged 10 to 18) with duration of 2 to 4 minutes were selected. There were a total of 170 fillers in these statements (from 14 to 37 in the statements of individual people). Over 70% efficiency and predictability of automatic detection of these disfluencies was achieved.

The procedures presented in the paper using the correlation coefficient can also be applied to find other types of disfluency, e.g. repetitions or stops. After building a sufficiently large database, the fillers can be adjusted to fully automatically detect the set type of disorder. The use of an automatic method in conjunction with therapy for a stuttering person can give us the opportunity to objectively assess the disorder, as well as to evaluate the progress of therapy.

REFERENCES

Alharbia, S., Hasana, M., Simonsa, A. J. H., Brumfitt, S., & Green, P. (2020). Sequence labeling to detect stuttering events in read speech. *Computer Speech & Language, 62,* 101052. http://doi.org/10.1016/j.csl.2019.101052

Bloodstein, O. (1995). *A handbook on stuttering.* Singular Publishing Group, Inc.

Czyżewski, A., Kaczmarek, A., & Kostek, B. (2003). Intelligent processing of stuttered speech. *Journal of Intelligent Inform. Systems,* 143–171.

Howell, P., & Sackin, S. J. (1995). Automatic recognition of repetitions and prolongations in stuttered speech, Stuttering. *Proceedings of the First World Congress on Fluency Disorders* (pp. 372–374). Munich.

Howell, P., Sackin, S. J., Glenn, K., & Au-Yeung, J. (1997). *Automatic stuttering frequency counts, Speech Motor Production and Fluency Disorders*. Elsevier.

Kuniszyk-Jóźkowiak, W., Dzieńkowski, M., Smołka E., & Suszyński, W. (2003). Computer Diagnosis and Therapy of Stuttering. *Structures – Waves – Human Health, VIII*(2), 133–144.

Kuniszyk-Jóźkowiak, W., Smołka, E., & Suszyński, W. (2001). Acoustical characteristics alteration in persons who stutter resulting from therapy. *Structures-Waves-Biomedical Engineering, X(2),* 57–68.

Kuniszyk-Jóźkowiak, W., Smołka, E., Dzieńkowski, M., & Suszyński W. (2004). Computer therapy of speech non-fluency with automatic adaptation of individual person's difficulties. *Structures-Waves-Human Health, VIII*(2), 63–70.

Moore, B. C. J., & Glasberg, B. R. (1983). Suggested formulae for calculating auditory-filter banwidths and excitation patterns. *The Journal of the Acoustical Society of America*, *74*, 750–753.

Moore, B. C. J., Peters, R. W., & Glasberg, B. R. (1990). Auditory filters shapes at low center frequencies. *The Journal of the Acoustical Society of America*, *88*, 132–149.

Smołka, E., Kuniszyk-Jóźkowiak, W., Suszyński, W., & Dzieńkowski, M. (2003). Speech syllabic structure extraction with application of Kohonen network. *Annales Informatica Universitatis Mariae Curie-Skłodowska, AI 1*,125–131.

Stromsta, C. (1993). The nature and management of stuttering. *Proceedings Abstracta, Congressus XVIII* (pp. 16–18). Societatis Phoniatricae Europaeae, Praga.

Suszyński, W., Kuniszyk-Józkowiak, W., Smolka, E., & Dzienkowski, M. (2003). Automatic Recognition of Nasals Prolongations in the Speech of Persons who Stutter. *Structures-Waves-Human Health, XII*(2), 175–184.

Suszyński, W., Kuniszyk-Jóźkowiak, W., Smołka, E., & Dzieńkowski, M. (2003). Prolongation detection with application of fuzzy logic. *Annales Informatica Universitatis Mariae Curie-Skłodowska, AI 1*, 133–140.

Suszyński, W., Kuniszyk-Jóźkowiak, W., Smołka, E., & Dzieńkowski, M. (2005). Speech disfluency detection with correlative method. *Annales Informatica Universitatis Mariae Curie-Skłodowska, AI 3*, 131–138.

Świetlicka, I., Kuniszyk-Jóźkowiak, W., & Smołka, E. (2013). Hierarchical ANN system for stuttering identification. *Computer Speech & Language*, *27*(1), 228–242. https://doi.org/10.1016/j.csl.2012.05.003

Wingate, M. E. (2002). *Foundation of stuttering*. Academic Press.

Wiśniewski, M, Kuniszyk-Jóźkowiak, W., Smołka, E., & Suszyński, W. (2010). Improved Approach to Automatic Detection of Speech Disorders Based the Hidden Markov Models Approach. *Journal of Medical Informatics & Technologies*, *15*, 145–152. http://doi.org/10.1007/978-3-540-75175-5_56

Wiśniewski, M., & Kuniszyk-Jóźkowiak, W. (2015). Automatic detection of stuttering in a speech. *Journal of Medical Informatics & Technologies*, *24*, 31–37.

Applied Computer Science, vol. 17, no. 4, pp. 55–74
doi: 10.23743/acs-2021-29

Submitted: 2021-09-17 | Revised: 2021-11-09 | Accepted: 2021-12-06

Keywords: track prediction, deep learning, education

Rowell HERNANDEZ [0000-0002-8748-6271]*,
Robert ATIENZA [0000-0001-7351-0325]**

CAREER TRACK PREDICTION USING DEEP LEARNING MODEL BASED ON DISCRETE SERIES OF QUANTITATIVE CLASSIFICATION

Abstract

In this paper, a career track recommender system was proposed using Deep Neural Network model. This study aims to assist guidance counselors in guiding their students in the selection of a suitable career track. It is because a lot of Junior High school students experienced track uncertainty and there are instances of shifting to another program after learning they are not suited for the chosen track or course in college. In dealing with the selection of the best student attributes that will help in the creation of the predictive model, the feature engineering technique is used to remove the irrelevant features that can affect the performance of the DNN model. The study covers 1500 students from the first to the third batch of the K-12 curriculum, and their grades from 11 subjects, sex, age, number of siblings, parent's income, and academic strand were used as attributes to predict their academic strand in Senior High School. The efficiency and accuracy of the algorithm depend upon the correctness and quality of the collected student's data. The result of the study shows that the DNN algorithm performs reasonably well in predicting the academic strand of students with a prediction accuracy of 83.11%. Also, the work of guidance counselors became more efficient in handling students' concerns just by using the proposed system. It is concluded that the recommender system serves as a decision tool for counselors in guiding their students to determine which Senior High School track is suitable for students with the utilization of the DNN model.

1. INTRODUCTION

The K-to-12 education system is a newly implemented educational system in the Philippines, and it is the last country in Asia to implement this curriculum (Abarro, 2016). The implementation of the new curriculum in the Philippines gives additional two years of education for Senior High School to prepare the students and empower them to confidently join the labor market (Roy Montebon, 2014) even if students don't choose to go to college. The Department of Education opted for gradual implementation of K-12 that will provide Filipino students sufficient time for mastery of concepts and skills so that they will be ready for tertiary education when the time comes.

* Batangas State University (Computer Science), Philippines, National Research Council of the Philippines – Engineering and Industrial Research, rowell.hernandez@g.batstate-u.edu.ph
** Batangas State University (Information Technology), Philippines, johnrobert.atienza@g.batstate-u.edu.ph

One of the aims of the K-12 curriculum is to increase the competitiveness of students' graduates in the country, and with the current situation in the K-12 assessment, proper counseling should be conducted in the selection of appropriate careers (Gorad, Zalte, Nandi & Nayak, 2017). The guidance counselor should also determine if the student can work in a particular field. Also, it is essential for teachers to periodically monitor their students' current performance on skills to make sure that the students are learning and improving every school year. In the K-12 curriculum career tracks were categorized into four namely: academic track, sports track, arts, and design tracks, and the technical-vocational track (Gestiada, Nazareno & Roxas-Villanueva, 2017).

Moreover, there are various career options available in each track, and many career opportunities in every field offered to the students (Laguador, 2014), and this exposed students with the various factors associated with career choices. That is why student's performance evaluation plays an important role in determining the strengths and weaknesses of the student before choosing the appropriate career track.

However, in the field of educational data mining, the most important topics are about the academic performance and the socio-demographic data of the students. The students' performance is an essential part of every learning institution, and the final grades of students are generally used to evaluate students' performance. The final grades of students are based on assessment marks, major exam scores, course structure, and other extracurricular activities provided by the institution (Bin Mat, Buniyamin, Arsad & Kassim, 2014). These data such as students' grades and socio-demographic data can be extracted to provide meaningful information about the students' status on each course or subject. It is also important that these data be utilized in helping students become aware of their strengths and what needs to be improved to achieve academic success and decision making.

Hence, with the available data on education, a career track recommendation system can be used to assist guidance counselors and students in the selection of appropriate career track. This recommender system could apply methods and techniques from statistics, data mining techniques, and neural networks to the problem of making a suitable recommendation of career tracks for senior high school students. The students should be satisfied with the services offered by the institution and career guidance because students are considered as the customers. Such a system with the utilization of neural networks can also help increase student satisfaction and maintain a joint relationship between the school and the students by helping the students in the selection of the right career track in Senior High School that matches with his skills and abilities.

A lot of the junior high school students experienced track uncertainty and were left confused on making decisions about choosing which career track at the senior high school level is applicable and suits the students. There are also instances that students shift to another program after learning they are not suited for their chosen course in college. According to Bin Mat et al. (2014), many students made wrong decisions on selecting a career due to a lack of experience, support, and advice from friends, parents, relatives, and teachers, or career counseling. The number of students and the number of choices is also growing in public schools, making it difficult for advisers to spend more time counseling each student due to workload (Goyal, Kukreja, Agarwal & Khanna, 2015). Also, the problem of student retention in Senior High School, especially in higher education, can further give rise to low student contentment wherein students are shifting from one course to another and dropping out (Razak et al., 2014). Dropout incidence is a bigger concern at the collegiate level than

at the secondary and elementary level due to individual-related problems such as poor academic performance, and health issues. Students are shifting to another program or strand after learning they are not suited on their chosen track or strand. This results to a waste of budget allocated by the government in State Universities for free tuition fees.

Furthermore, it is very important to help the students to increase their awareness (Durosaro & Nuhu, 2019) about the career tracks that are suitable for them, and not to just pick any course that they want (Asif, Merceron & Pathan, 2015). The students of junior high school need to select one career track from these four categories under the K-12 curriculum before entering senior high school. Career guidance should be delivered in several ways to every student so that it can help students to be more aware of selecting appropriate career tracks based on the student's overall academic performance.

The student's performance evaluation will help in the determination of the student's strengths and weaknesses before choosing a career track. It is only high time to develop a system that will assist guidance counselors in completing the performance evaluation of students. Moreover, it could be beneficial for the teachers and counselors to have a decision tool that empirically shows the academic performance analysis of students (Grewal & Kaur, 2015), and recommended career track for the students. The developed method can also provide the students with quality and convenient support services, and with the utilization of certain algorithms, career-track-related decisions will be supported and deduced.

This paper aims to design and develop a neural network-based career track for junior high school students using Deep Neural Network (DNN). The algorithms that were used in this study will give an update about the progress of students in each subject every grading period. While providing grades and early predictions on the future academic performance of students, the utilization of DNN in the developed system will classify the prediction results of academic achievement to determine which Senior High School track is suitable for the students. This developed system will provide students with recommendations in choosing a career track that is appropriate to their skills and abilities. The results of the prediction will help the teachers and guidance counselor to interpret information and apply it to their students' situation in selecting career track by using the DNN algorithm approach.

2. LITERATURE REVIEW

In this section, literature related to student academic performance prediction and classification is reviewed.

The researchers discussed the implementation of the additional two years in the Philippine high school system. As part of the program, students are set to choose one track from ten academic strands. With several factors to consider, the selection of a career path may be difficult for a student. The researcher proposed a study that aims to create a tool that will guide students in choosing a particular career track using Social Cognitive Career Theory (SCCT) and the analytic hierarchy process (AHP). To identify the factors in considering the selection of career, the researchers used the SCCT, whereas the AHP was used in ranking the career tracks according to these factors. Evaluation of the tool in terms of design, navigation, and utility was also conducted on more than 150 Grade 10 students using pilot testing (Gestiada, Nazareno & Roxas-Villanueva, 2017).

Meanwhile, the selection of the right course in formative years is a very important decision as students' future depends on this one decision. The student by himself is not mature enough to decide his early life. The selection of wrong courses means a mismatch between student aptitude, capability, and personal interest. Also, the faculty or parents have neither the required knowledge nor experience. Since there is no other reliable source generally available that can guide a student towards the most suitable direction, the recommender system has been evolved to provide students' guidance in selecting a right course. This paper proposes feasible predictions for students' course selection based on final marks and choice of job interest (Grewal & Kaur, 2015). To find structure and relationship within the data, the clustering technique was used, and it is said the technique can work on unsupervised data.

A proposed career recommender system using Fuzzy logic was presented which aims to help not only the guidance counselor but most especially the Senior High School students to guide them in considering numerous factors associated with their decision on what career they will pursue. In dealing with choosing the student attributes from numerous factors, a feature selection technique is appropriate to use to remove irrelevant features that affect the performance of the proposed fuzzy-based system. In this paper, different filter methods are used to select the best attributes (Natividad, Gerardo & Medina, 2019; Razak et al., 2014; Qamhieh, Sammaneh & Demaidi, 2020; Sulaiman, Tamizi, Shamsudin & Azmi, 2019). After selecting the best attributes, these are now used as crisp inputs. The result of the experiment shows a reasonable result for making decisions (Alzhrani & Algethami, 2019). It is concluded that the proposed career recommender system for students assists students in their career decision. The proposed system for students is also very timely and will be one of the significant researches works in the new era of the education system in the Philippines.

In another study of Okubu et al. (2017) about students' performance prediction, the researchers show a method that will predict students' final class marks using a Recurrent Neural Network. For this purpose, the learning logs from 937 students who attended one of six courses by two teachers were collected. Nine kinds of learning logs are selected as the input of the RNN. The researchers carefully examine the prediction of final class marks, where the training data and test data are the logs of courses conducted in 2015 and 2016, respectively (Tai-Nghe, Drumond, Krohn-Grimberghe & Schmidt-Thieme, 2010) The study shows that observing the weight values of the trained RNN helps identify the important learning activities when it comes to obtaining a specific final class mark.

According to the study of Rafanan et al. (2020), the artificial neural network approach can be used in predicting the career strand of incoming senior high school students. The K-12 program gives additional two years in the students' basic education, and these ancillary years allow senior high school students to take courses under the core curriculum and the track of choice. Each student must select one track to pursue that can equip him/her with skills to prepare for the future. Prediction of choice of a career track in senior high school is advantageous for educational institutions since it gives insights that can help them develop vital programs beneficial for students learning in school. In this study, the applied artificial neural network (ANN) to predict the career strand based on the students' grades in five major subjects. Different ANN models have been considered and compared. In training and testing the models, a sample of 293 student data information was used (Nazareno et al., 2019). The highest accuracy recorded among all the models was 74.1%.

A neural network called the Deep Neural Network model was proposed in another study that shows students which class category it belongs to. This study provides knowledge to the institution so that proper remedy can be offered to the potential failing students. A comparison with existing machine learning algorithm which uses the same dataset with the proposed model (Bendangnuksung & Prabu, 2018). With larger dataset records and features, a DNN can achieve higher accuracy and will outperform the other machine learning algorithm (Vijayalakshmi & Venkatachalapathy, 2019). Two hidden layers are implemented Relu and Soft-Max activation function. The prediction of students failing is effective, with an estimated 85% accuracy, and outperforms other machine learning algorithms inaccuracy.

Moreover, the study of Piad et al. (2016) predicted the employability of IT graduates using nine variables. First, different classification algorithms in data mining were tested making logistic regression with an accuracy of 78.4 is implemented. Based on logistic regression analysis, three academic variables directly affect; IT_Core, IT_Professional, and Gender identified as significant predictors for employability. The data were collected based on the five-year profiles of 515 students randomly selected at the placement office tracer study.

Several kinds of research in the educational field that involve Data mining techniques are (Hamsa, Indiradevi & Kizhakkethottam, 2016) rapidly increasing. The researcher applied Data Mining techniques in the field of education that aims to discover hidden knowledge and patterns about students' performance. This work aims to develop students' academic performance prediction model, for the Bachelor and Master degree students in Computer Science and Electronics and Communication streams using two selected classification methods; Decision Tree and Fuzzy Genetic Algorithm. The resultant prediction model can be used to identify students' performance for each subject (Jauhari & Supianto, 2019).

Thereby, the lecturers can classify students and take early action to improve their performance. Systematic approaches can be taken to improve the performance with time. Due to early prediction and solutions being done, better results can be expected in final exams (Hasan et al., 2018). The students can be able to view their academic information and updates in school. Moreover, the results from the decision tree algorithm made more students at risk class, which makes lecturers decision to take more care of those students. Results from the fuzzy logic algorithm give more past students considering those who are in between risk and safe, to a safe state that gives students mental satisfaction.

Furthermore, another study about the use of a classification model for predicting the suitable study track for school students was presented by researchers. Researchers said that one of the most important issues in academic life is to assign students to the right track when they arrive in the end of the basic education stage (Al-Radaideh, Ananbeh & Al-Shawakfa, 2011). The main issue in the selection of an academic track in basic Jordanian schools is the lack of useful knowledge for students to support their planning. A decision tree classification model was developed to determine which track is suitable for each student. There are set of classification rules that were extracted from the decision tree to predict (Al-Barrak & Al-Razgan, 2016) and classify the class label for each student. A confusion matrix is built to evaluate the model (Varade & Thankanchan, 2021) where the 10-fold Cross Validation method was used for accurate estimation of the model. The overall accuracy of the model was 87.9% where 218 students were correctly classified out of the 248 students.

In the polytechnic system, a student must take the elective subjects at least three subjects to complete their study. The researchers decided to use the Decision Tree method for predicting students' performance in the elective subjects. The elective subjects were chosen based on their interest and first come first serve. The results of the final examination for elective subjects affect the future of the students, and it is important to predict whether the students will pass or fail in the final examination. To obtain the necessary information about students' profiles, the literature survey was used. The researcher of this paper uses data mining which is the decision tree method for the prediction of the student's performance in each elective subject (Sulaiman, 2020; Khasanah & Harwati, 2017). This research is focused on the ICT students who select DBM3033 as an elective subject, and the two phases involved were preprocessing the data and mining the data. The RapidMiner software is used in the data mining process.

Classification technique is applied for decision tree method. Some attributes are collected from the students' database record to predict the final grade in DBM3033. From the experiments, the average training accuracy is 71.11% and the accuracy for the testing data is 77.50%. Therefore, it looks like the accuracy is still in the good range. The research findings showed that students whose results are weak in both SPM Mathematics and DBM1033 are predicted as fail in final examination for DBM3033 (Sulaiman, Shibghatullah & Rahman, 2017).

According to the researches, the data mining techniques can be applied to predict and analyze students' academic performance based on students' academic records and forum participation. This paper explained that educational institutions can use educational data mining for extensive analysis of students' characteristics (Rizvi, Rienties & Khoja, 2019; Abu Zohair, 2019; Mhetre & Nagar, 2018). Three different data mining classification algorithms (Naïve Bayes, Neural Network, and Decision Tree) were used and applied in the dataset. The prediction performance of three classifiers was measured and compared. It was observed that the Naïve Bayes classifier outperforms the other two classifiers by achieving an overall prediction of 86%. While the two classifiers achieved 82% and 79% for the DT classifier. This study help teachers to improve student academic performance (Mueen, Zafar & Manzoor, 2016).

3. PROPOSED WORK AND METHODOLOGY

3.1. Proposed approach

This section, it shows the processes to the methodology used in this study. This includes data collection, data pre-processing, data cleaning, model building, model evaluation, and interpretation. The first and foremost step is to identify the data and collect the dataset required for the study.

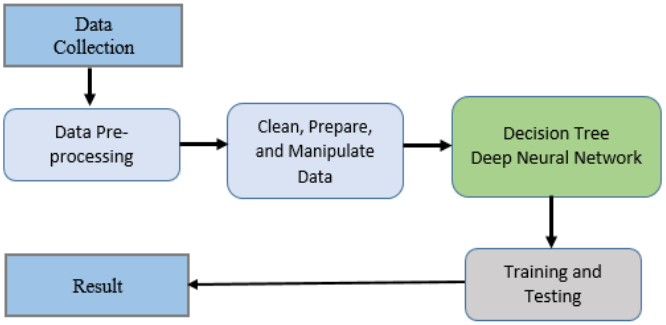


Fig. 1. System Model Architecture

Figure 1 shows the system model architecture. The class marks of students in each subject and their socio-demographic data were collected. After the required data is gathered, the raw data must be pre-processed to achieve the proper format and remove the unnecessary or inconsistent data that may affect the performance of the model. The efficiency and accuracy of the algorithms depend on the correctness and quality of the data collected. The next step is to use the two well-known classification techniques which are Decision tree and Deep Neural Network. These two techniques were used to classify students' academic performance according to the Senior High School academic track. Moreover, the researcher discussed the data pre-processing and strategies performed in this study.

3.2. Methodologies

3.2.1. Data Collection

Table 1 shows the initial data collected from the student during the data collection process. The grades of student for each grading period and their final average in each subject were collected and used in the creation of the DNN and Decision tree model.

Tab. 1. Collected Data from Students

No	Attributes
1	Student Name
2	Track
3	Sex
4	Age
5	No. of Siblings
6	Salary Bracket of Parents
7	Occupation of Mother
8	Occupation of Father
9	Place of Birth
10	Name of Region
11	Subject Grades

The gathered response from the students about their socio-demographic data contained the following: Name, Age, Sex, Place of Birth, Name of Region, Number of Siblings, Occupation of Mother, Occupation of Father, Family Salary Bracket, and Academic Track. In the collection of data, the researcher ensured that ethical procedures were properly performed before, during, and after the actual data gathering. Class scores of students and socio-demographic data were collected through an online structured questionnaire. Concentrating on the required data, the names of the respondents were removed so that no student can be identified in this study.

3.2.2. Data Cleaning

The collected datasets were downloaded and organized in an excel sheet. During the data pre-processing, the researcher cleaned and replaced the students' data with appropriate data. The cleaning of data was carefully applied to make it suitable for the model. All null values found were removed so that it would not be difficult for the predictive model to learn.

3.2.3. Normalization

Normalization is an essential step in data pre-processing because it transforms data in a way that the data have similar distributions. In this paper, the researcher ensured that input requirements are prepared for Deep Neural Network and Decision tree algorithms. The student's attributes with text data were converted into numerical values. Then, normalization is applied by scaling each attribute and grades of students in each subject between 0 and 1. In this study, the min-max type of normalization is adopted in which it scales every feature value between its minimum (0) and maximum (1). The validation accuracy of the model increases after applying the normalization.

3.2.4. Data Preprocessing

Feature engineering is used in the selection or creation of variables in a dataset to improve the prediction results. It is a process of transforming collected data into features that will act as inputs to the machine learning models.

The two important parts of this data pre-processing were variable transformation and feature creation. The features were created by extracting the data in a variable, removing the unused features (such as names of students, occupation of parents, and region). Fine-tuning of hyper-parameters was also applied in this study, and it was used to find the best combination of parameters for the predictive model. The data-preprocessing procedures were conducted in this study after the grades and socio-demographic data were gathered. The steps and strategies are as follows.

[a] To begin with, all the grades in eleven subjects were normalized such that it less one. This means that the grades in each subject are all divided by 100 so each grade varied within the same range from 0 to 1.

[b] While all categorical data were converted into indexes. For example, Male was assigned as 1, and Female was assigned as 0.

[c] The Place of Birth has a high correlation to the Region, and it is because the student's birthplace is where their family stays.

[d] The occupation of the mother and the occupation of the father has a high correlation with the salary bracket of the family. The income of their parents is defined by the occupation and contributes to the Salary Bracket of the Family unless they also have other sources of income.

[e] In addition, the region was dropped since most of the respondents all come from the same region which is CALABARZON 4A.

[f] Place of Birth could be a factor for a student's upbringing and choice of academic track/strand but inclusion in this would require more data from different schools in the country to prove the claim.

[g] Furthermore, the salary bracket of the family is taken over the occupation of the mother and the father since this has few numerical categories compared to the number of occupations taken from the survey, and the salary bracket is more reasonable to use in the model as discussed in [a].

Tab. 2. Student related attributes

No.	Attributes	Category	Frequency
1	**Student ID**	Student ID No.	1500
2	**Track**	STEM	500
		HUMSS	500
		ABM	500
3	**Sex**	Male	465
		Female	1035
4	**Age**	18yrs old	171
		19yrs old	502
		20yrs old	465
		21yrs old	337
		22yrs old	21
		23yrs old	4
5	**No. of Siblings**	0	91
		1	324
		2	417
		3	300
		4	168
		5	107
		6	38
		7	22
		8	12
		9	13
		10	8
6	**Salary bracket of Parents**	1 – Poor (<9250)	30
		2 – Low Income (Between 9250–19040)	891
		3 – Low Middle Income (19040–38080)	432
		4 – Middle Income (38040–66640)	83
		5 – Upper Middle Income (66640–114240)	58
		6 – Upper Income (114240–190000)	6
7	**Subject Grades**	Filipino, English, Mathematics, Science, Social Science (AP), Technology and Livelihood Education, Edukasyon sa Pagpapakatao, Music, Arts, Physical Education, and Health	1500

Table 2 shows the final attributes and descriptions of student records. Each student's socio-demographic and academic record had the following attributes.

The dataset is comprised of 1,500 undergraduate degree students and then were categorized into three academic tracks (STEM, HUMSS, and ABM) which consisted of 500 (33.34%), 500 (33.33%), and 500 (33.33%) students. In table 2, the final dataset or attributes were applied in the creation of the DNN and Decision Tree model.

3.2.5. Made Learning Model

For the model construction, a deep neural network and decision tree method has been used.

Studen No	FIL	ENG	MATH	SCI	AP	TLE	EP	MUSIC	ARTS	PE	HEALTH	Age	Sex	Siblings	SalaryBracket	STRAND
502	78.75	76.8125	76.25	77.375	77.6875	76.9375	81.0625	81	80.8125	80.125	79.8125	21	F	4	Low Income	HUMSS
503	77.5	77.5625	78.5	77.5625	78.1875	77.5	82.1875	79.625	80.25	79.0625	79.875	20	F	1	Low Income	HUMSS
238	84	83.125	83.25	83.75	83.125	83.625	82.5	87.0625	87	86.6875	86.0625	21	F	2	Low Middle In	ABM
198	81.9375	84.3125	84.8125	84.9375	84.0625	84.125	82.5625	90.875	90.25	90.5625	89.8125	19	F	2	Low Middle In	ABM
181	85.25	84.0625	83.375	84.5625	80.875	81	82.9375	85.125	85.5625	86.1875	86.9375	21	M	3	Low Income	ABM
491	85.25	84.0625	83.375	84.5625	80.875	81	82.9375	85.125	85.5625	86.1875	86.9375	19	M	2	Low Middle In	ABM
832	86	86.0625	84.625	85.875	84.4375	84.9375	83.3125	85.4375	85.6875	86.375	86.6875	19	F	2	Low Income	HUMSS
199	83.0625	84	84.4375	82.9375	83.1875	84.0625	83.375	90.875	90.8125	89.9375	90.25	19	F	2	Low Middle In	ABM
228	85.5625	84.875	85.8125	87.125	86.125	84.8125	83.4375	86.625	86.875	87.125	88.25	21	F	5	Low Income	ABM

Fig. 2. Sample of input variables for the first 9 students

Figure 2 shows the format of the final dataset used in the training and testing of the model. From the dataset collected 70 percent of the data was utilized for training, and 30 percent of the data is reserved for testing. The DNN and Decision Tree algorithms were applied to the dataset and results are noted and observed.

3.2.6. Deep Neural Network and Decision Tree

A Deep Neural Network is described as a kind of machine learning with multiple hidden layers between the input and output layers. The neurons in the neural network are used for the processing of information, which is interconnected to sense the propagation of signals. These networks of neurons become useful when applied in solving problems of prediction and classification (Oancea, Dragoescu & Ciucu, 2013). The architecture of a deep neural network which consists of an input layer, hidden layer, and output layer is shown in Figure 3. The number of layers and number of neurons in each layer were also discussed in the next page.

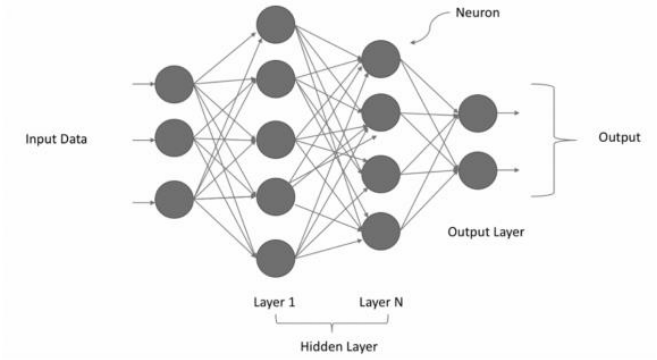


Fig. 3. Deep Neural Network Architecture

In this study, the Deep Neural Network is employed in the model to perform the classification (Sarvepalli, 2015) and prediction of a career track for students. According to Yi et al. (2017), DNN aims at transforming the data towards a more abstract and innovative element. All neurons of input layers are fed to the neurons of hidden layers to process the input, while outputs are obtained from the output layer. The hyperparameters used in this study were sigmoid activation function, epochs, input layer consists of 8 nodes, hidden layers which consists of 8 nodes, while the final layer consists of 3 nodes. These hyperparameters obtained were used by the researcher to establish the DNN predictive model.

The sigmoid activation function was used because it exists between 1 to 0, and this activation function is especially utilized for the model that can predict the probability as an output. Since the probability of anything exists only between the range of 0 and 1, sigmoid is the right choice. While the 4000 epochs for the DNN model are determined during the training phase, the final number of epochs used was taken due to high and consistent accuracy. Moreover, the number of layers was more on experimentation, the first layer is usually the number of features the researcher wants to feed the model, the number of the second layer should be at least greater than the first layer then the addition of hidden layers.

Finally, the final layer in Deep Neural Network algorithm consists of three nodes i.e., [ABM], [HUMSS], and [STEM], and each has a percentage for the recommendation of the academic strand. The highest percentage the student obtained from the three strands was the one recommended for the student, but students are free to choose which career track to pursue in senior high school. The guidance counselor can use the result to guide and evaluate their students in the selection of career track.

Moreover, the Decision Tree classifier is also used for the prediction of the academic track of students and compared the result of validation accuracy to DNN results. A decision tree is a flow-chart-like tree structure, where each internal node is denoted by rectangles, and leaf nodes are denoted by ovals. All internal nodes have two or more child nodes. All internal nodes contain splits, which test the value of an expression of the attributes. Arcs from an internal node to its children are labeled with distinct outcomes of the test. A class label is associated with each leaf node (Pal, 2011). Moreover, the decision tree algorithm is beneficial in data mining when it comes to handling a variety of textual, numeric, and nominal types of documents (Li & Zhang, 2011). This method can be used in datasets with a large number of errors and missing values. In this study, the maximum depth of the tree is three. One of the focuses of this study is to determine which Senior High School track is suitable for the students. The system provides the students a recommendation in choosing a career track based on their academic performance in each subject. The result of the prediction of which career track to choose can assist guidance counselors to interpret information and apply it to their students' situations in the selection of career track using the Deep Learning algorithm. This study helps students have a better insight into their future performance and make informed decisions by recommending a suitable strand under an academic track.

3.2.7. Tools and Techniques

The Python programming language was used in creating and testing of the DNN and Decision Tree model, while the Google Colaboratory serves as the Integrated Development Environment which helps in analyzing and visualizing the dataset. Python is a powerful

general-purpose programming language. It is a mature and rapidly growing platform for scientific investigation and numerical computation, and Python hosts a large number of open source libraries as well as almost all general purpose machine learning libraries that can be used to train the deep learning and decision tree models. Moreover, Colab Notebooks are Jupyter notebooks that are useful in for generating and presenting data science projects in an interactive manner, and it supports variety of programming languages such as Python.

Table 3 shows the output of the Deep Neural Network model and the prediction of students' academic track.

Tab. 3. Final Output of the DNN Model for the first 11 students

Student No	ABM	STEM	HUMSS
0	67.65	1.47	30.88
1	28.82	64.75	6.43
2	28.82	64.75	6.43
3	75.69	23.2	1.1
4	28.82	64.75	6.43
5	8.86	5.14	86
6	28.82	64.75	6.43
7	8.86	5.14	86
8	28.82	64.75	6.43
9	75.69	23.2	1.1
10	67.65	1.47	30.88

The results of prediction were classified according to the Senior High School academic track and is presented in this table. The exact parameters used in this DNN model were grades of student in each subject, sex, age, number of siblings, and salary bracket of parents.

3.3. Evaluation metrics

To evaluate the prediction model 5-fold cross-validation was used, and the percentage split method is applied. In the percentage split method, the dataset training set (70%) is used to train the model while the remaining 30% is for testing the model. A comparison of validation accuracy between Neural Network and Decision Tree was tested to ensure the highest prediction or classification could be achieved in this study. Moreover, the training set was divided into 5 disjoint sets of approximately equal size. The 5-fold cross-validation of data is an iterative process in which the process was repeated five times and then the accuracy of the predictive model is computed.

To compare the prediction model, the Root Means Square Root was used. This evaluation metric is a standard deviation of the errors that occur when a prediction of the academic track is made on a collected dataset. The $\boldsymbol{RMSE}$ evaluation metric is the same as MSE (Mean Squared Error) but the value of its root is considered while determining the accuracy of the predictive model as in the following equation:

$$RMSE = \sqrt{\left(\frac{\sum(\hat{Y}_i - Y_i)^2}{n}\right)} \quad (1)$$

where: $\hat{Y}_i$ and Y_i are the predicted and targeted values, and n is the total number of records.

The smaller the $RMSE$ values the better as it is an indication of a good prediction of the target values. Generally, the simpler and easier model to interpret is preferred if the compared predictive models have no significant difference (Obsie & Adem, 2018).

4. RESULTS AND DISCUSSIONS

Table 4 illustrates the performance of DNN and Decision Tree methods after the testing process. As the results indicate, the two predictive models performed reasonably well in classifying and predicting the academic strand of the student.

Tab. 4. Prediction result of two algorithms

Prediction Methods	Accuracy %	MSE	RMSE
Decision Tree	0.7889	0.5844	0.7644
Deep Neural Network	0.8311	0.4555	0.6749

Table 4 shows that the Deep Neural Network method produced the more accurate prediction results, in which 0.8311 accuracy, 0.4555 MSE, and 0.6749 RMSE values were obtained with the 5-fold cross-validation test. While Decision tree method obtained a 0.7889 accuracy, 0.5844 MSE, and 0.7644 RMSE respectively. The parameters used in DNN and Decision tree model were grades in each subject, sex, age, number of siblings, and salary bracket of parents. The input layer in DNN model consists of 8 nodes, and hidden layers consists of 8 nodes, while the final layer consists of 3 nodes. The Decision Tree model has a maximum depth of three. Moreover, tree depth is the number of splits a tree can make before making a prediction.

Figure 4 shows the graph of the testing accuracy and loss from the predictive model. The loss in this graph is a value that represents the summation of errors in the predictive model. This measures how good or bad the model is performing. It is important to note that if the errors are high, then the loss will be high, which indicates that the predictive model does not perform well. Otherwise, the lower it is, the better the model works when it comes to predicting the academic strand of the students.

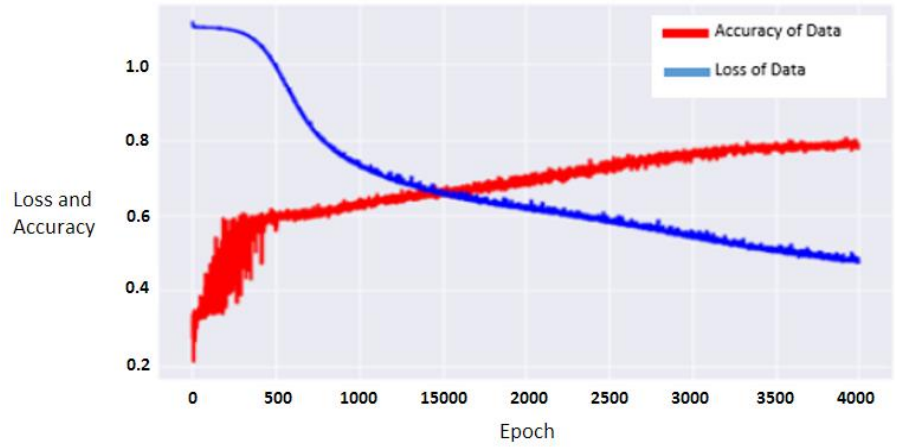


Fig. 4. Accuracy and Loss Graph

In this figure 4, the loss data is decreasing and we can see the expected behavior of the learning process even if it has slight ups and downs. The loss decreases over time, so the predicted model is learning. While accuracy in the graph describes the percentage of test data that are classified correctly, the higher it is, the better the predictive model becomes. With the proposed deep neural network, it was able to achieve an accuracy of 83.11% in predicting the academic strand of a student. Great accuracy with low loss means that the model made low errors on a few data and is considered as the best case.

A comparison-based study is also made to Nazareno et al.'s (2019) proposed model. The researchers used and suggested an Artificial Neural network model in predicting the career strand of incoming senior high school students. The parameters used in this proposed ANN model were grades in Filipino, English, Math, Science, and Technology and Livelihood Education. The DNN and Decision tree predictive models used in this study were compared with their ANN prediction accuracy. The prediction accuracy of the three models is calculated and recorded. Table 5 shows the result of the comparison of accuracy.

Tab. 5. Comparison of Accuracy Among Different Techniques

Classifier	Accuracy
Artificial Neural Network	74.1%
Decision Tree	78.89%
Deep Neural Network	83.11%

The table shows that the accuracy of our predictive model Deep Neural Network is 83.11%, while other techniques got an accuracy of 78.89% for Decision tree and 74.1% for Artificial Neural Network. Also, the ANN value is less than the decision tree value, which is why it can be concluded that DT and DNN work better than the other model when it comes to predicting performance and classifying students' academic strands. Moreover, the two predictive models (Decision Tree and Deep Neural Network) have been compared to one another using the ROC index performance measure.

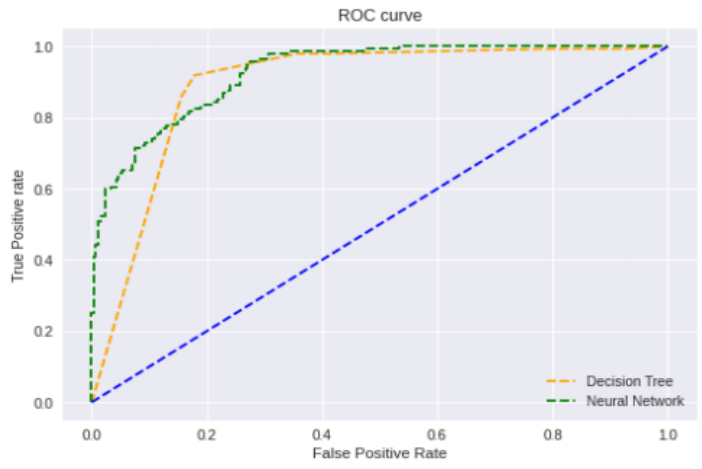


Fig. 5. DNN and DT ROC index

Figure 5 shows that DNN model has the highest ROC index that is equal to 0.9271. While the decision tree model got an accuracy of 79%. This means that the DNN model is much better than the DT model when it comes to predicting the performance or academic strand of students. In addition, the Deep Neural Network is more versatile compared to the Decision Tree model in most cases. The DNN provides percentages which is more useful for the recommender setup, where the Decision Tree gives a specific academic strand of choice. This means that DNN performed well in predicting the academic strand of students using the same attributes. Moreover, all the data used during the training phase came from the real-life values which actual student grades and their response from the surveys. The researcher split the test set into 20-80 and 30-70 portions into 5-folds where the researcher determined which parameters and features to use that provides the highest accuracy.

4.1. Confusion Matrix

The researcher used the confusion matrix to determine the accuracy of strand prediction. This summarizes the number of correct and incorrect predictions made by the model in a tabular format. The actual value and predicted value are indicated in the table below. The performance of the model is also evaluated using the accuracy, precision, and recall performance metrics. Accuracy is a proportion of the total number of correct predictions of a strand, while Precision indicates the proportion of correct positive observations. The recall is a proportion of positives correctly predicted as positive.

Tab. 6. Confusion Matrix for DNN

	ABM (Actual)	STEM (Actual)	HUMSS (Actual)	Classification Overall	Precision
ABM (Predicted)	108	3	36	147	73.469%
STEM (Predicted)	3	135	9	147	91.837%
HUMSS(Predicted)	7	18	131	156	83.974%
Truth overall	118	156	176	450	
Recall	91.525%	86.538%	74.432%		

Table 6, shows the actual and classifier results. There are 147 students from ABM and STEM strands, and it is observed that 108 students were predicted correctly under the ABM strand, while the STEM strand predicted 135 students out of 147 students under the STEM strand. Moreover, there are 156 students in the HUMSS strand and 131 students are predicted correctly. The overall accuracy of the DNN model is 83.11%.

Tab. 7. Interpretation of Values in Cohen's Kappa Statistics

Values	Interpretation
Smaller than 0.00	Poor Agreement
0.00 to 0.20	Slight Agreement
0.21 to 0.40	Fair Agreement
0.41 to 0.60	Moderate Agreement
0.61 to 0.80	Substantial Agreement
0.81 to 1.00	Almost Perfect Agreement

Cohen's Kappa Statistic is also used to measure the inter-rater agreement for categorical items (ABM, STEM, HUMSS). The Cohen's Kappa value for this study has a 0.746% value which means that there is a substantial agreement as shown in Table 7.

4.2. Comparative Analysis of Significant Parameters

The original parameters or attributes used in the model were Grades, Age, Siblings, Sex, and Salary Bracket of parents. This achieved an accuracy of 83.11% in the prediction of strands for the DNN model. The table shows the comparative analysis of significant parameters and their corresponding accuracy.

Table 8 below shows 10 model cases having different combinations of parameters and determine which parameter made a significant effect on the performance of the model used in the system. Case 9 and 10 got the lowest accuracy of 59.11% and 62.32% with the combination of the following attributes: Grades in 11 subjects, age, sex, number of siblings, salary bracket of parents, region, and subject groupings. Comparing Case 3 and 4, Case 3 has an accuracy of 82.44% with the following attributes: Grades, Age, and Sex. While Case 4 has achieved an overall accuracy of 80.66% with attributes Grades, Age, Sex, and Number of Siblings.

Another comparison was made between Case 3 and 5, and it shows that predictive accuracy increased after removing the number of siblings and replacing it with the salary bracket of parents attribute in Case 5 with an accuracy of 82.66%. Moreover, it is observed that the most important attributes were the grades of the student, and the combination of Age, Sex, Number of Siblings, and Salary Bracket of Parents with an accuracy of 83.11%. The difference of accuracy results between the two cases 1 (Original) and 6 is only a small percentage.

Tab. 8. Comparative analysis of best attributes

	Grades in 11 subjects	Age	Sex	No. of siblings	Salary Bracket of Family	Region	Subject Groupings (4 groups)	Accuracy
Original	✓	✓	✓	✓	✓	X	X	83.11%
Case 2	✓	X		X	X	X	X	79.33%
Case 3	✓	✓	✓	X	X	X	X	82.44%
Case 4	✓	✓	✓	✓	X	X	X	80.66%
Case 5	✓	✓	✓	X	✓	X	X	82.66%
Case 6	✓	X	✓	X	✓	X	X	83.09%
Case 7	X	X		X	X	X	✓	63.55%
Case 8	✓	X	✓	✓	✓	X	X	81.55%
Case 9	✓	✓	✓	✓	✓	✓	✓	59.11%
Case 10	✓	✓	✓	✓	✓	✓	X	62.32%

5. CONCLUSION AND FUTURE WORKS

The researcher successfully collected the grades and socio-demographic data of the students following the strict compliance and requirements by the University's Data Privacy Rules. The two models (Decision Tree and Deep Neural Network) have been compared to one another using the ROC index performance measure, and classification accuracy. The DNN model has the highest ROC index that is equal to 0.9271 and has an accuracy of 83%. While the decision tree model got an accuracy of 79%. To solve the mismatch of students in selecting academic strand and help students in deciding which career track to pursue in Senior high school. The grades in each subject and socio-demographic data of students are the attributes used in the Deep Neural Network model. This is a new development in terms of larger parameters as compared to the existing studies. This study is the first to utilize more data to train the model. The study can be more processed if the dataset can be increased from different geographic locations of the Philippines which can probably make the Machine Learning model better than what the researcher has achieved.

This study shows that it is possible to predict and classify the academic performance of the students. It is also concluded that the DNN technique can be used efficiently in classifying students' academic performance in junior high school. With the utilization of an algorithm, the work of the guidance counselor became more efficient in handling students' concerns and providing a remedy to the student who is undecided in selecting a career track.

For future works, the development of an online career track recommender system was recommended so that counselors can have a decision tool in guiding their students in the determination of which Senior High School track is to pursue and suitable for them. Having an online viewing of grades and recommended career track for students can be helpful to the students who are undecided in career track selection. Furthermore, the development of an online recommender system with the utilization of an algorithm serves as an awakening factor for education agencies to be in line with the government's view of globalization and competitiveness in the 21st century or today's information age.

Acknowledgment

The authors would like to acknowledge the Philippine Statistical Research and Training Institute (PSRTI) in providing financial support in attaining the objectives of the study, the National Research Council of the Philippines, and the Digital Transformation Center of STEER HUB Batangas State University for allowing us to use their laboratory.

REFERENCES

Abarro, J. O. (2016). Factors Affecting Career Track and Strand Choices of Grade 9 Students in the Division of Antipolo and Rizal, Philippines. *International Journal of Scientific and Research Publications*, *6*(6), 51–2250.

Abu Zohair, L. M. (2019). Prediction of Student's performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education*, *16*(1), 27. https://doi.org/10.1186/s41239-019-0160-3

Al-Barrak, M. A., & Al-Razgan, M. (2016). Predicting Students Final GPA Using Decision Trees: A Case Study. *International Journal of Information and Education Technology*, *6*(7), 528–533. https://doi.org/10.7763/ijiet.2016.v6.745

Al-Radaideh, Q. A., Ananbeh, A. A., & Al-Shawakfa, E. M. (2011). A classification model for predicting the suitable study track for school students. *IJRRAS*, *8*(2), 18788963.

Alzhrani, N., & Algethami, H. (2019). Fuzzy-Based Recommendation System for University Major Selection. In *Proceedings of the 11th International Joint Conference on Computational Intelligence – Volume 1: FCTA* (pp. 317–324). Vienna, Austria. https://doi.org/10.5220/0008071803170324

Asif, R., Merceron, A., & Pathan, M. K. (2015). Investigating performance of students: A longitudinal study. *LAK '15: Proceedings of the Fifth International Conference on Learning Analytics And Knowledge* (pp. 108–112). ACM International Conference Proceeding Series. https://doi.org/10.1145/2723576.2723579

Bendangnuksung, & Prabu, D. (2018). Students' Performance Prediction Using Deep Neural Network. *International Journal of Applied Engineering Research*, *13*(2), 1171–1176.

Bin Mat, U., Buniyamin, N., Arsad, P. M., & Kassim, R. A. (2014). An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention. *2013 IEEE 5th International Conference on Engineering Education: Aligning Engineering Education with Industrial Needs for Nation Development (ICEED)* (pp. 126–130). IEEE. https://doi.org/10.1109/ICEED.2013.6908316

Durosaro, I. A., & Nuhu, M. A. (2012). An evaluation of the relevance of career choice to school subject selection among school going adolescents in ondo state. *Asian Journal Of Management Sciences And Education*, *1*(2), 140–145.

Gestiada, G., Nazareno, A., & Roxas-Villanueva, R. M. (2017). Development of a senior high school career decision tool based on social cognitive career theory. *Philippine Journal of Science*, *146*(4), 445-455.

Gorad, N., Zalte, I., Nandi, A., & Nayak, D. (2017). Career Counseling using Data Mining. *International Journal of Engineering Science and Computing*, *7*(4), 10271–10274.

Goyal, P., Kukreja, T., Agarwal, A., & Khanna, N. (2015). Narrowing awareness gap by using e-learning tools for counselling university entrants. *2015 International Conference on Advances in Computer Engineering and Applications* (pp. 847–851). IEEE. https://doi.org/10.1109/ICACEA.2015.7164822

Grewal, D.S., & Kaur, K. (2015). Developing an Intelligent Recommendation System for Course Selection by Students for Graduate Courses. *Business and Economics Journal*, *7*(2), 1000209. https://doi.org/10.4172/2151-6219.1000209

Hamsa, H., Indiradevi, S., & Kizhakkethottam, J. J. (2016). Student Academic Performance Prediction Model Using Decision Tree and Fuzzy Genetic Algorithm. *Procedia Technology*, *25*, 326–332. https://doi.org/10.1016/j.protcy.2016.08.114

Hasan, R., Palaniappan, S., Raziff, A. R. A., Mahmood, S., & Sarker, K. U. (2018). Student Academic Performance Prediction by using Decision Tree Algorithm. *2018 4th International Conference on Computer and Information Sciences: Revolutionising Digital Landscape for Sustainable Smart Society (ICCOINS)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCOINS.2018.8510600

Jauhari, F., & Supianto, A. A. (2019). Building student's performance decision tree classifier using boosting algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(3), 1298–1304. https://doi.org/10.11591/ijeecs.v14.i3.pp1298-1304

Khasanah, A. U., & Harwati. (2017). A Comparative Study to Predict Student's Performance Using Educational Data Mining Techniques. *IOP Conference Series: Materials Science and Engineering*, *215*(1), 012036. https://doi.org/10.1088/1757-899X/215/1/012036

Laguador, J. (2014). Examination of Influence and Intention towards Lyceum of the Philippines University and Career Choice of General Engineering Students. *International Journal of Management Sciences*, *3*(11), 847–855.

Li, L., & Zhang, X. (2010). Study of data mining algorithm based on decision tree. *2010 International Conference on Computer Design and Applications* (V1-155-V1-158). IEEE. https://doi.org/10.1109/ICCDA.2010.5541172

Mhetre, V., & Nagar, M. (2018). Classification based data mining algorithms to predict slow, average and fast learners in educational system using WEKA. *Proceedings of the International Conference on Computing Methodologies and Communication (ICCMC 2017)* (pp. 475–479). IEEE. https://doi.org/10.1109/ICCMC.2017.8282735

Mueen, A., Zafar, B., & Manzoor, U. (2016). Modeling and Predicting Students' Academic Performance Using Data Mining Techniques. *International Journal of Modern Education and Computer Science*, *8*(11), 36–42. https://doi.org/10.5815/ijmecs.2016.11.05

Natividad, M. C. B., Gerardo, B. D., & Medina, R. P. (2019). A fuzzy-based career recommender system for senior high school students in K to 12 education. *IOP Conference Series: Materials Science and Engineering*, *482*(1), 012025. https://doi.org/10.1088/1757-899X/482/1/012025

Nazareno, A. L., Lopez, M. J. F., Gestiada, G. A., Martinez, M. P., & Roxas-Villanueva, R. M. (2019). An artificial neural network approach in predicting career strand of incoming senior high school students. *Journal of Physics: Conference Series*, *1245*(1), 012005. https://doi.org/10.1088/1742-6596/1245/1/012005

Oancea, B., Dragoescu, R., & Ciucu, S. (2013). Predicting students ' results in higher education using neural networks. *International Conference on Applied Information and Communication Technology* (pp. 190–193). Jelgava (Latvia).

Obsie, E. Y., & Adem, S. A. (2018). Prediction of Student Academic Performance using Neural Network, Linear Regression and Support Vector Regression: A Case Study. *International Journal of Computer Applications*, *180*(40), 975–8887.

Okubo, F., Yamashita, T., Shimada, A., & Konomi, S. (2017). Students' performance prediction using data of multiple courses by recurrent neural network. *Proceedings of the 25th International Conference on Computers in Education, ICCE 2017 - Main Conference Proceedings* (pp. 439–444). Asia-Pacific Society for Computers in Education.

Pal, S. (2011). PER-07: A prediction for performance improvement using classification. India – Chapter III. *Student Related Variables*, *9*(4).

Piad, K. C., Dumlao, M., Ballera, M. A., & Ambat, S. C. (2016). Predicting IT employability using data mining techniques. *2016 3rd International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)* (pp. 26–30). IEEE. https://doi.org/10.1109/DIPDMWC.2016.7529358

Qamhieh, M., Sammaneh, H., & Demaidi, M. N. (2020). PCRS: Personalized Career-Path Recommender System for Engineering Students. *IEEE Access*, *8*, 214039–214049. https://doi.org/10.1109/ACCESS.2020.3040338

Rafanan, R. J. L., De Guzman, C. Y., & Rogayan, D. V. (2020). Pursuing stem careers: Perspectives of senior high school students. *Participatory Educational Research*, *7*(3), 38–58. https://doi.org/10.17275/per.20.34.7.3

Razak, T. R., Hashim, M. A., Noor, N. M., Halim, I. H. A., & Shamsul, N. F. F. (2014). Career path recommendation system for UiTM Perlis students using fuzzy logic. *2014 5th International Conference on Intelligent and Advanced Systems: Technological Convergence for Sustainable Future* (pp. 1–5). IEEE. https://doi.org/10.1109/ICIAS.2014.6869553

Rizvi, S., Rienties, B., & Khoja, S. A. (2019). The role of demographics in online learning; A decision tree based approach. *Computers and Education*, *137*(January), 32–47. https://doi.org/10.1016/j.compedu.2019.04.001

Roy Montebon, D. T. (2014). K12 Science Program in the Philippines: Student Perception on its Implementation. *International Journal of Education and Research*, *2*(12), 153–164.

Sarvepalli, S. S. K. (2015). *Deep Learning in Neural Networks: The science behind an Artificial Brain*. https://doi.org/10.13140/RG.2.2.22512.71682

Sulaiman, M. S., Tamizi, A. A., Shamsudin, M. R., & Azmi, A. (2019). Course recommendation system using fuzzy logic approach. *Indonesian Journal of Electrical Engineering and Computer Science*, *17*(1), 365–371. https://doi.org/10.11591/ijeecs.v17.i1.pp365-371

Sulaiman, S. (2020). Prediction Students ' Performance in Elective Subject Using Decision Tree Method. *Journal of Asian Islamic Higher Institutions (JAIH)*, *5*(1).

Sulaiman, S., Shibghatullah, A. S., & Rahman, N. A. (2017). Prediction of students' performance in elective subject using data mining techniques. *Proceedings of Mechanical Engineering Research Day 2017* (pp. 222–224).

Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., & Schmidt-Thieme, L. (2010). Recommender system for predicting student performance. *Procedia Computer Science*, *1*(2), 2811–2819. https://doi.org/10.1016/j.procs.2010.08.006

Varade, R. V., & Thankanchan, B. (2021). Academic Performance Prediction of Undergraduate Students using Decision Tree Algorithm. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology*, *13*(SUP 1), 97–100. https://doi.org/10.18090/samriddhi.v13is1.22

Vijayalakshmi, V., & Venkatachalapathy, K. (2019). Comparison of Predicting Student's Performance using Machine Learning Algorithms. *International Journal of Intelligent Systems and Applications*, *11*(12), 34–45. https://doi.org/10.5815/ijisa.2019.12.04

Yi, H., Shiyu, S., Duan, X., & Chen, Z. (2017). A study on Deep Neural Networks framework. *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016* (pp. 1519–1522). IEEE. https://doi.org/10.1109/IMCEC.2016.7867471

Applied Computer Science, vol. 17, no. 4, pp. 75–83
doi: 10.23743/acs-2021-30

Submitted: 2021-10-08 | Revised: 2021-12-10 | Accepted: 2021-12-20

Keywords: keystroke dynamics analysis, machine learning, neural network, supervised learning, classification problem

Nataliya SHABLIY [0000-0002-1125-4757]*, *Serhii LUPENKO* [0000-0002-6559-0721]*, *Nadiia LUTSYK* [0000-0002-0361-6471]*, *Oleh YASNIY* [0000-0002-9820-9093]*, *Olha MALYSHEVSKA* [0000-0003-0180-2112]**

KEYSTROKE DYNAMICS ANALYSIS USING MACHINE LEARNING METHODS

Abstract

The primary objective of the paper was to determine the user based on its keystroke dynamics using the methods of machine learning. Such kind of a problem can be formulated as a classification task. To solve this task, four methods of supervised machine learning were employed, namely, logistic regression, support vector machines, random forest, and neural network. Each of three users typed the same word that had 7 symbols 600 times. The row of the dataset consists of 7 values that are the time period during which the particular key was pressed. The ground truth values are the user id. Before the application of machine learning classification methods, the features were transformed to z-score. The classification metrics were obtained for each applied method. The following parameters were determined: precision, recall, f1-score, support, prediction, and area under the receiver operating characteristic curve (AUC). The obtained AUC score was quite high. The lowest AUC score equal to 0.928 was achieved in the case of linear regression classifier. The highest AUC score was in the case of neural network classifier. The method of support vector machines and random forest showed slightly lower results as compared with neural network method. The same pattern is true for precision, recall and F1-score. Nevertheless, the obtained classification metrics are quite high in every case. Therefore, the methods of machine learning can be efficiently used to classify the user based on keystroke patterns. The most recommended method to solve such kind of a problem is neural network.

1. INTRODUCTION

It is hard to imagine modern world without different novel technologies. Therefore, the task of data protection is of high importance.

The authentication problem is as follows. Two parties are talking with each other, and one or both want to send their identity to the other (Gebrie & Abie, 2017). Authentication is

* Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Systems and Networks Department, Ternopil, Ukraine, natalinash@gmail.com, serhii.lupenko@gmail.com, lutsyk.nadiia@gmail.com, oleh.yasniy@gmail.com
** Ivano-Frankivsk National Medical University, Department of Hygiene and Ecology, Ivano-Frankivsk, Ukraine, o16r02@gmail.com

the process of verifying the physical identity of a person and digital identity of a computer. User authentication is a cornerstone of any information system.

The principles, that are the basis of identification and authentication methods, can be divided into four groups (Gebrie & Abie, 2017):

- traditional password protection;
- verification of physical parameters of human (fingerprints, iris scanning, etc.) (Dhir et al., 2010);
- assessment of psycho-physical parameters;
- estimation of user information interests and dynamics of its change.

The password-based authentication is widely used in identity verification (Hwang, Lee & Cho, 2009). Nevertheless, it becomes unsafe when a password is obtained by third-party. Keystroke dynamics-based authentication (KDA) was invented that propose increased security (Gaines, Lisowski, Press & Shapiro, 1980). KDA is based on the fact that a user's keystroke patterns repeat themselves and are unique (Umphress & Williams, 1985). It can be employed in internet banking, ATM, and smartphones, which require high level of security. It is possible to add fingerprint, iris, and voice to the traditional password-based authentication (Jain, Bolle, & Pankanti, 2006; Dhir et al., 2010). Also, KDA needs special equipment and requires several actions of user (Ru & Eloff, 1997; Monrose, Reiter, & Wetzel, 2002).

It is clear that any biometric is not the best recognition method in all cases and its selection is specific for certain application. A comparison of features on seven factors is provided in Table 1 (Jain, Ross & Prabhakar, 2004).

Some systems require user to provide a card before it can get access the data of the system. The examples of such cards are credit cards, debit cards, cash-machine cards. Cards can have either a magnetic strip or a computer chip. Cards containing a computer chip are also known as smart cards. With this system, the user must provide such card before the machine will allow that person to access any information. With a key-lock system, a person must unlock the computer to get access to the system. Most PCs have a key-lock installed that allows the authorized user to lock out the keyboard. When the system is locked, keyboard input is not recognized. Cards and keys can be lost, stolen, or forged. Also, the key-locks on PCs can be disabled if a person can remove the case of the machine. This radical method is generally not necessary, since most PC locks use the same type of key. If a person has a computer with a key-lock, then it is possible that his or her key can open or close the lock on another unauthorized computer (Fischer, Halibozek & Walters, 2019).

PINs, passwords, and digital signatures are compatible with any computer system. PINs work in conjunction with various types of card systems. With this system, one inserts a card and then enters the PIN, a security number known only to the user. Passwords are special words required to access a computer system. Companies should require passwords to contain at least eight characters that could be any combination of special symbols, capital and lowercase letters, and numbers. Easily guessed or obvious passwords should not be employed in practice. Finally, the company may assign passwords to employees that are random combination of numbers, letters, or special symbols. If the system requires a higher degree of security, then a password should only be used once. Those are so called one time passwords (Fischer, Halibozek & Walters, 2019).

Tab. 1. Features of most common biometrics characteristics (Jain, Ross & Prabhakar, 2004)

Biometric characteristic	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention
Facial thermogram	H	H	L	H	M	H	L
Hand vein	M	M	M	M	M	M	L
Gait	M	L	L	H	L	H	M
Keystroke	L	L	L	M	L	M	M
Odor	H	H	H	L	L	M	L
Ear	M	M	H	M	M	H	M
Hand geometry	M	M	M	H	M	M	M
Fingerprint	M	H	H	M	H	M	M
Face	H	L	M	H	L	H	H
Retina	H	H	M	L	H	L	L
Iris	H	H	H	M	H	L	L
Palmprint	M	H	H	M	H	M	M
Voice	M	L	L	M	L	H	H
Signature	L	L	L	H	L	H	H
DNA	H	H	H	L	H	L	L

Digital signatures system uses a public/private key system. The sender creates the signature with a public key, and the receiver reads it with a second, private key. The two largest drawbacks of the mentioned above systems are associated with passwords and PINs (Fischer, Halibozek & Walters,2019).

Passwords can be guessed. Users tend to use real words or dates (their name, birth date, friends' or children's names, user initials, ids, and so on). Some users even do not replace the default initial password. PINs and passwords are often written down by users in places that can be easily accessed by others (Fischer, Halibozek & Walters,2019).

Biometrics methods are based on measuring individual body features. Fingerprints, hand geometry, retinal characteristics, voice recognition, keystroke dynamics, signature dynamics are common ways to identify authorized users. The computer compares the item being scanned with a copy of the item stored in the computer's memory. If the compared items match, the computer allows access, or denies otherwise (Fischer, Halibozek & Walters, 2019).

The one of the most important issues with KDA is in the fact that keystroke patterns from unauthorized users are not available while training classifier (Hwang, Lee & Cho, 2009). Therefore, it is very hard to build binary classifier. This can be eliminated using novelty detection framework. The idea of novelty detection method is to identify the novel or abnormal patterns that occur in a large amount of normal data (Miljković, 2010). Novelty or outlier is a pattern in the data that signifies unexpected behavior. The aim of novelty detection is to determine abnormal system behaviors which differs from the normal state of a system (Chandola, Banerjee & Kumar 2009; Markou & Singh, 2003, Miljković, 2010).

In the study (Hwang, Lee & Cho, 2009), there was proposed to use artificial rhythms and tempo cues to provide consistency and uniqueness of typing patterns. Different novelty detectors were built based on various artificial rhythms and/or tempo cues. It was shown show that artificial rhythms and tempo cues improve authentication accuracies and can be implemented in real world authentication systems.

However, the approach with binary classifier does not take into account the patterns of another user trying to impersonate the one, its password it is typing. To overcome this limitation, instead of binary classifier, proposed in novelty detection approach, in the current study, the multiclass classifier was employed that enables authentication of specific user that enters the password. This allows adding extended security to the computer system.

There was performed the analysis of keystroke dynamics based on methods of machine learning. The time of delay while pressing the keyboard buttons was measured and was used to predict the user of the system among the known list of authorized persons. In this case, the classification task was solved.

2. METHODS

KDA was performed using the following supervised methods of machine learning: logistic regression, support vector machines, random forest, and neural networks.

Logistic regression, despite its name, is a classification model rather than regression model (Subasi, 2020). Logistic regression is method that allows determining the probability of a discrete outcome given an input variable. The most common logistic regression model deals with binary outcome; something that can take two values such as true/false, yes/no, 1/0, etc. Multinomial logistic regression is a model where there are more than two possible discrete outcomes. Logistic regression is used for classification tasks (Edgar & Manz, 2017). Python Scikit-learn module contains an optimized logistic regression implementation, which allows multiclass classification (Raschka, 2017).

Support vector machine (SVM) works as follows (Vaibhaw, Sarraf & Pattnaik, 2020). A hyperplane or a set of hyperplanes is created, that separate the feature vectors into several classes. It selects the hyperplane which is at the maximum distance from the nearest training samples. SVM determines the hyperplane with the maximal margin by mapping input data into high-dimensional space. SVM also employs regularization to prevent artifacts. Nonlinear SVM have a nonlinear decision boundary that is based on kernel function.

Random forest (RF) models are machine learning algorithms that make predictions by combining outcomes from a set of regression decision trees (Williams et al., 2020). Each tree is built independently and is based on a random vector sampled from the input data, with all the trees in the forest having the same distribution. The predictions from the forests are averaged using bootstrap aggregation and random feature selection. RF models are robust predictors for both small sample sizes and high dimensional data (Biau & Scornet, 2016).

Neural networks (NN) are computing systems inspired by the biological neural networks (Kohonen, 1982), which learn to solve tasks by considering examples without being programmed with any specific rules. The neural networks were applied to solve the variety of classification tasks in (Alyamani & Yasniy, 2020; Dewi & Utomo, 2021; Sridharan et al., 2021). The training of neural networks can be achieved in several ways, using, for instance, the approach called particle swarm optimization that was employed in (Al-Awad, Abboud & Al-Rawi, 2021).

3. RESULTS AND DISCUSSION

Each of three users typed the same word that has 7 symbols 600 times. The row of the dataset consists of 7 values that are the time period during which the particular key was pressed. The ground truth values are the user id (either 1, 2 or 3). The task was to predict the user based on the typed word.

Data normalization scales the feature values to make them belong to the same interval, and, therefore, have the same importance. Because most machine learning algorithms produce better models when the data are normalized, the numerical data should be normalized or standardized before classification. There are three most commonly employed normalization techniques: z-score normalization, min-max normalization, and normalization by decimal scaling (Javaheri, Sepehri & Teimourpour, 2013). For this study the z-score normalization was applied. The data were normalized using its mean and standard deviation. After the preprocessing, all features have a mean of zero and a standard deviation of one. For each variable, this was performed by subtracting the mean of the variable and dividing by the standard deviation.

The dataset was divided into two unequal parts, namely, the training set and the testing set. The testing set contained 33% of the dataset, while the training set consisted of remaining 67% of the entire dataset.

Four methods of supervised learning were employed: logistic regression, support vector machines, random forest, and neural networks, similarly to (Alyamani & Yasniy, 2020). For each method, the normalized confusion matrices were obtained. Fig. 1 shows the normalized confusion matrices, built by means of machine learning methods for the mentioned above dataset.

The obtained results are based on the modern methods of machine learning and main postulates of statistics and probability theory.

The confusion matrix is commonly used measure that is employed while solving classification tasks. It can be equally applied to binary classification as well as for multiclass classification task. Confusion matrices contain counts from predicted and actual values.

To obtain the normalized confusion matrix, the corresponding row of original confusion matrix was divided into number of dataset samples that were created by each user. In this study, this number was equal to 600.

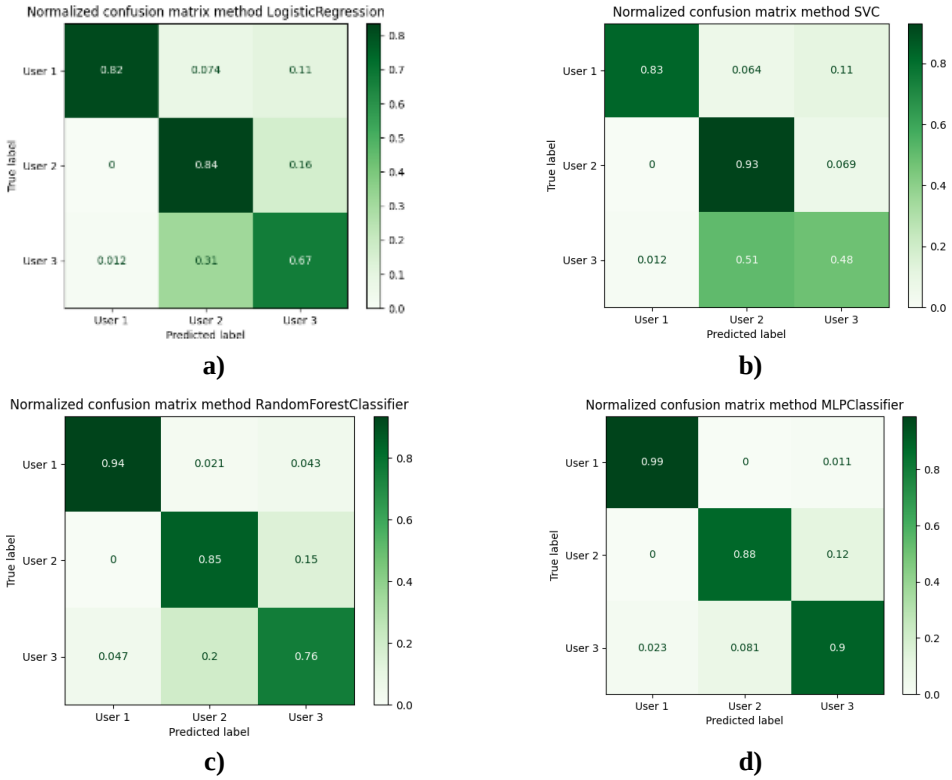


Fig. 1. Normalized confusion matrices obtained by various methods of machine learning: a) Logistic regression, b) Support vector machines, c) Random forest, d) Multilayer Perceptron (Neural network)

Using neural network approach, user 1 was detected in 99% of cases. The second place has user 3 with 90% of detection. The last place had user 2 with 88% of detection. In general, user 3 was misclassified most frequently as user 2 in the methods of support vector machines in around 51% of cases. The method of logistic regression classified 31% percent of user 3 samples as user 2. The same pattern is true for Random forest classifier with 20% of user 3 samples misclassified as user 2.

The classification metrics were obtained for each applied method. The following parameters were determined: precision, recall, f1-score, support, prediction, and area under the receiver operating characteristic curve (AUC). AUC provides a measure of performance across all possible classification thresholds. AUC takes value from [0, 1] (Bradley, 1997).

In case of neural network, its topology and hyperparameters were as follows: there were 3 hidden layers with numbers of neurons on each layer equal to ith element of the tuple (150, 10, 10), the employed algorithm was limited memory Broyden-Fletcher-Goldfarb-Shanno L-BFGS, the maximum number of iterations was equal to 1000000, the learning rate alpha was equal to 0.001.

Tab. 2. Logistic regression

Class	Precision	Recall	F1-score	Support	Predicted	AUC
1	0.987	0.819	0.895	94	78	0.984
2	0.740	0.836	0.785	116	131	0.906
3	0.667	0.674	0.671	86	87	0.863
Avg/Total	0.797	0.784	0.787	296	296	0.928

Tab. 3. Support vector classifier

Class	Precision	Recall	F1-score	Support	Predicted	AUC
1	0.987	0.829	0.901	94	79	0.989
2	0.683	0.931	0.788	116	158	0.903
3	0.694	0.476	0.565	86	59	0.841
Avg/Total	0.783	0.766	0.759	296	296	0.924

Tab. 4. Random forest

Class	Precision	Recall	F1-score	Support	Predicted	AUC
1	0.956	0.936	0.946	94	92	0.995
2	0.838	0.853	0.846	116	118	0.956
3	0.755	0.755	0.755	86	86	0.933
Avg/Total	0.852	0.851	0.851	296	296	0.968

Tab. 5. Neural network

Class	Precision	Recall	F1-score	Support	Predicted	AUC
1	0.978	0.989	0.984	94	95	0.985
2	0.935	0.879	0.906	116	109	0.971
3	0.836	0.895	0.865	86	92	0.948
Avg/Total	0.920	0.918	0.919	296	296	0.977

Tables 2–5 contain the classification metrics (precision, recall, F1-score, support, predicted, as well as AUC score for each class and for the dataset in total). The obtained AUC score is quite high. The lowest AUC score that was equal to 0.928 was achieved in the case of linear classifier. The highest AUC score was in the case of neural network classifier. The method of support vector machines and random forest showed slightly lower results as compared with neural network method. The same pattern is true for precision, recall and F1-score. Therefore, the methods of machine learning can be efficiently used to classify the user based on keystroke patterns. The best method that solved this task is neural network. Particularly, the proposed approach can be used in computer information systems to add another level of security and provide increased protection from potential intruders.

4. CONCLUSIONS

The task of users classification based on their keystrokes patterns was solved using the methods of supervised machine learning: logistic regression, support vector machines, random forest, and neural network. The multiclass classifier was built that allows determining the user based on its keystroke dynamics analysis with high accuracy. The method with highest classification score was neural network. The method with the lowest classification metrics was logistic regression. In general, the AUC score, obtained with each method, was more than 0.92. Therefore, such kind of task can be efficiently solved by means of machine learning approaches. This approach can be used in computer information systems to add another level of security and provide additional protection from potential intruders. In the future research, there can be used the extended dataset that includes data from a larger amount of users. Also, the hyperparameter optimization can be performed to increase the classification metrics.

REFERENCES

Al-Awad, N. A., Abboud, I. K., & Al-Rawi, M. F. (2021). Genetic Algorithm-PID controller for model order reduction pantographcatenary system. *Applied Computer Science, 17*(2), 28-39. https://doi.org/10.23743/acs-2021-11

Alyamani, A., & Yasniy, O. (2020). Classification of EEG signal by methods of machine learning. *Applied Computer Science*, *16*(4), 56-63. https://doi.org/10.23743/acs-2020-29

Biau, G., & Scornet, E. (2016). A Random Forest Guided Tour. *Test*, *25*(2), 197–227. https://doi.org/10.1007/s11749-016-0481-7

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Dewi, W., & Utomo, W. H. (2021). Plant classification based on leaf edges and leaf morphological veins using wavelet convolutional neural network. *Applied Computer Science*, *17*(1), 81–89. https://doi.org/10.23743/acs-2021-08

Dhir, Vijay, Singh, A., Kumar, R., & Singh, G. (2010). Biometric Recognition: A Modern Era For Security. *International Journal of Engineering Science and Technology*, *2*(8), 3364–80.

Edgar, T. W., & Manz, D. O. (2017). *Research Methods for Cyber Security*. Syngress.

Fischer, R. J., Halibozek, E. P., & Walters, D. C. (2019). Holistic Security Through the Application of Integrated Technology. *Introduction to Security, 2019*, 433–62. https://doi.org/10.1016/b978-0-12-805310-2.00017-2.

Gaines, R. S., Lisowski. W., Press, S. J., & Shapiro, N. (1980). *Authentication by Keystroke Timing*. The Rand Corporation.

Gebrie, M. T., & Abie, H. (2017). Risk-Based Adaptive Authentication for Internet of Things in Smart Home EHealth. *Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings (ECSA'17)* (pp. 102–108). Association for Computing Machinery. https://doi.org/10.1145/3129790.3129801

Hwang, S.-S., Lee H., & Cho, S. (2009). Improving Authentication Accuracy Using Artificial Rhythms and Cues for Keystroke Dynamics-Based Authentication. *Expert Systems with Applications*, *36*(7), 10649–56. https://doi.org/10.1016/j.eswa.2009.02.075

Jain, A. K., Bolle, R. M., & Pankanti, S. (2006). *Biometrics. Personal Identification in Networked Society*. Springer.

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An Introduction to Biometric Recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, *14*(1), 4-19.

Javaheri, S. H., Sepehri, M. M. & Teimourpour, B. (2013). Response Modeling in Direct Marketing. A Data Mining-Based Approach for Target Selection. Data Mining Applications with R (pp. 153-180). Elsevier Inc. https://doi.org/10.1016/B978-0-12-411511-8.00006-2

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69.

Markou, M., & Singh, S. (2003). Novelty detection: a review—part 1: statistical approaches. *Signal Processing, 83*(12), 2481–2497. https://doi.org/10.1016/j.sigpro.2003.07.018

Miljković, D. (2010). Review of novelty detection methods. *The 33rd International Convention MIPRO* (pp. 593-598). IEEE.

Monrose, F., Reiter, M. K., & Wetzel, S. (2002). Password Hardening Based on Keystroke Dynamics. *International Journal of Information Security*, *1*(2), 69–83. https://doi.org/10.1007/s102070100006

Raschka, S. (2017). *Python Machine Learning. Second edition.* Packt Publishing Ltd.

Ru, W.G., & Eloff, J.H. (1997). Enhanced Password Authentication through Fuzzy Logic. *IEEE Expert, 12*, 38-45.

Sridharan, M., Rani Arulanandam, D. C., Chinnasamy, R. K., Thimmanna, S., & Dhandapani, S. (2021). Recognition of font and tamil letter in images using deep learning. *Applied Computer Science, 17*(2), 90–99. https://doi.org/10.23743/acs-2021-15

Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python*. Academic Press.

Umphress, D., & Williams, G. (1985). Identity verification through keyboard characteristics. *International Journal of Man-Machine Studies*, *23*(3), 263–273. https://doi.org/10.1016/S0020-7373(85)80036-5

Vaibhaw, Sarraf, J., & Pattnaik, P.K. (2020). Brain–Computer Interfaces and Their Applications. *An Industrial IoT Approach for Pharmaceutical Industry Growth*, *2*, 31-54. https://doi.org/10.1016/b978-0-12-821326-1.00002-4

Williams, B., Halloin, C., Löbel, W., Finklea, F., Lipke, E., Zweigerdt, R., & Cremaschi, S. (2020). Data-Driven Model Development for Cardiomyocyte Production Experimental Failure Prediction. *Computer Aided Chemical Engineering*, *48*, 1639-1644. https://doi.org/10.1016/B978-0-12-823377-1.50274-3

Applied Computer Science, vol. 17, no. 4, pp. 84–99
doi: 10.23743/acs-2021-31

Submitted: 2021-11-03 | Revised: 2021-12-14 | Accepted: 2021-12-21

Keywords: Industry 4.0, CPS, IoT, machine monitoring

Jarosław ZUBRZYCKI [0000-0002-7454-8090]*, *Antoni ŚWIĆ* [0000-0003-0405-4009]*, *Łukasz SOBASZEK* [0000-0003-1298-2438]*, *Juraj KOVAC* [0000-0002-7793-9564]**, *Ruzena KRALIKOVA* [0000-0002-9231-7886]***, *Robert JENCIK*****, *Natalia SMIDOVA* [0000-0002-6511-4397]***, *Polyxeni ARAPI* [0000-0003-0009-6041]******, *Peter DULENCIN*******, *Jozef HOMZA*********

CYBER-PHYSICAL SYSTEMS TECHNOLOGIES AS A KEY FACTOR IN THE PROCESS OF INDUSTRY 4.0 AND SMART MANUFACTURING DEVELOPMENT

Abstract

The continuous development of production processes is currently observed in the fourth industrial revolution, where the key place is the digital transformation of production is known as Industry 4.0. The main technologies in the context of Industry 4.0 consist Cyber-Physical Systems (CPS) and Internet of Things (IoT), which create the capabilities needed for smart factories. Implementation of CPS solutions result in new possibilities creation – mainly in areas such as remote diagnosis, remote services, remote control, condition monitoring, etc. In this paper, authors indicated the importance of Cyber-Physical Systems in the process of the Industry 4.0 and the Smart Manufacturing development. Firstly, the basic information about Cyber-Physical Production Systems were outlined. Then, the alternative definitions and different authors view of the problem were discussed. Secondly, the conceptual model of Cybernetic Physical Production System was presented. Moreover, the case study of proposed solution implementation in the real manufacturing process was presented. The key stage of the verification concerned the obtained data analysis and results discussion.

* Lublin University of Technology, Lublin, Poland, j.zubrzycki@pollub.pl, a.swic@pollub.pl, l.sobaszek@pollub.pl
** Slovak Academy of Sciences, Bratislava, Slovakia, juraj.kovac@tuke.sk
*** Technical University of Kosice, Kosice, Slovakia, ruzena.kralikova@tuke.sk, natalia.smidova@tuke.sk
**** Manex s.r.o, Čaňa, Slovakia, robert.jencik@manex.sk
***** Technical University of Crete, Chania, Greece
****** Spojená škola Juraja Henischa, Bardejov, Slovakia, peter.dulencin@gmail.com, homzaj@gmail.com

1. INTRODUCTION

The key goal of Industry 4.0 (I4.0) is to be faster and increase production efficiency. Industry 4.0 combines a large number of new technologies to create value. The main technologies in the context of Industry 4.0 are Cyber-Physical Systems (CPS) and the Internet of Things (IoT). This approach is considered as a key enabling technology in the Fourth Industrial Revolution (i-SCOOP, 2021).

Cyber-Physical Systems use modern control systems, have embedded software systems and dispose of an Internet address to connect and be addressed via IoT. This way, products and means of production get networked and can "communicate", enabling new ways of production, value creation, and real-time optimization. Cyber-Physical Production Systems create the capabilities needed for Smart Factories (Harrison, Vera & Ahmad, 2016).

In the context of Industry 4.0 (mechanics, engineering, etc.) Cyber-Physical Systems are seen as the next step in the development of continuous production improvement through integration, interaction and communication (Onik, Kim & Yang, 2019). Looking at Industry 4.0 as the next new stage in the organization and control of the value chain during the product life cycle, mechanical systems began, mechatronics and adaptronics were introduced, and Cyber-Physical Systems are now beginning.

Cyber-Physical Systems essentially enable us to make industrial systems capable to communicate and network them, which then adds to existing manufacturing possibilities. They result to new possibilities in areas such as remote diagnosis, remote services, remote control, condition monitoring, systems health monitoring and so forth (Ratchev, 2017).

2. SOME VIEWS ON UNDERSTANDING CYBER-PHYSICAL SYSTEMS

1.1. Definitions of CPS

To understand Industry 4.0, it is necessary to introduce the following keyword "Cyber-Physical Systems" (CPS) which are the core of this topic. "Cyber Physical Systems" are intelligent embedded systems, a combination of electronics and software, which are connected to the real world through sensors and actuators, and are also connected to each other and to the Internet. Thus, the physical world merges with a virtual world to a cyberspace, which is, according to its definition, a combination of digitalized data, creating a universe of information and communication connected through the internet.

In the same way, there are Cyber-Physical Production System (CPPS) dedicated to the industrial field. They collect physical values like temperature, dimensions, displacement, pressure, force, etc. via different kind of sensors. Thanks to their computing capabilities, they can process this data with specific algorithms, for for example for predictive maintenance (Yasniy, Pyndus, Iasnii & Lapusta, 2017), and transfer them e.g. to a MES.

Cyber-physical systems before Industry 4.0: In the original definitions, going back over a decade, IP addresses where not specifically mentioned in Cyber-Physical Systems. In 2008, Professor Edward A. Lee from the University of California, Berkeley, defined Cyber-Physical Systems as follows: "Cyber-Physical Systems are integrations of computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa" (Ratchev, 2017).

The term “Cyber-Physical System” was originally coined by Ellen Gill in 2006. CPS is a category of embedded system. It is often called a next-generation computing system that uses intelligent computing techniques associated with the physical world and computing units. CPS can interact with real systems through calculations, communication and controls. The interaction of computational and physical units leads to advanced implementations of the Internet of Things. IoT and CPS are designed to support real-time applications that can manage many sets of environmental data (Sabella, 2018). In other words, CPS is a combination of digital control and the physical environment. The basic scheme of CPS is shown in Fig.1.

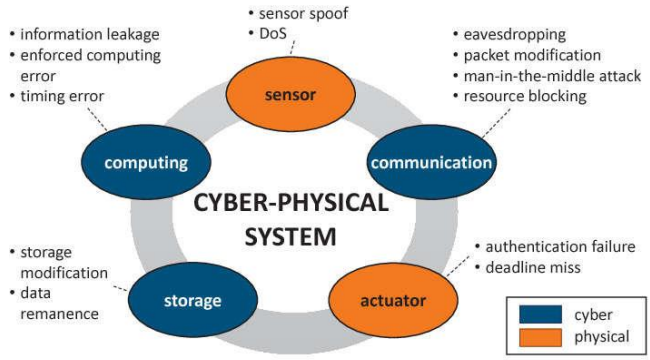


Fig. 1. Basic scheme of CPS

The Cyber-Physical System consists of cybernetic components and physical components, therefore we call it the cyber-physical system. CPS is based on a computer information processing system that is built into a product, such as an automobile, airplane, machine tool, or other device. This computer system interacts with the physical environment through sensors and actuators (Harrison, Vera & Ahmad, 2016).

These embedded systems are no longer separate, sharing their data through communication networks such as the Internet with cloud computing, where data from many embedded systems can be collected and processed (ADDI-DATA, 2015). This creates a system of systems. The collected data can be processed automatically or via the HMI – Human Machine Interface (Fig. 2).

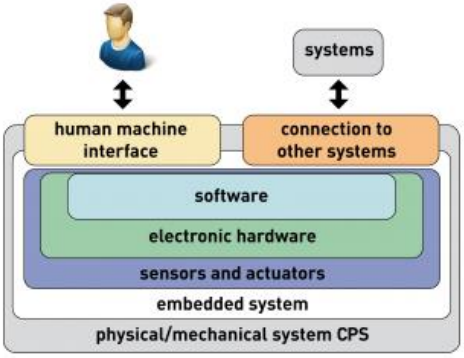


Fig. 2. CPS integrated subsystems (ADDI-DATA, 2015)

In (Schuh et al., 2014) CPS are defined by as cooperating systems, having a decentralized control, resulting from the fusion between the real world and the virtual world, having autonomous behaviors and dependent on the context in which they are, being able to constitute in systems of systems with other CPS and leading a deep collaboration with the human. For this, embedded software in CPS uses sensors and actuators, connect to each other and to human operators by communicating via interfaces, and have storage and data processing capabilities from the sensors or the network (Strang & Anderl, 2014).

The recent one, suggested by (Monostori, 2014), allows a clear synthesis of the various aspects of this large concept, coupling in addition the notion of services with CPS : "Cyber-Physical Systems are systems of collaborating computational entities which are in intensive connection with the surrounding physical world and its on-going processes, providing and using, at the same time, data-accessing and data-processing services available on the internet". To do so, embedded software in CPS uses sensors and actuators, connect with each other and with humans communicating via standard interfaces, and have abilities of storage and processing of data coming from sensors or from the network (Strang & Anderl, 2014). This interconnection of systems, as stated by (Gengarle et al., 2013), derives from the fact that a CPS encompasses together control, computation but also communication devices.

CPS is an intersection, not a union, of that which is considered virtual to that which is physical. It is no longer enough to separately understand, develop, manage and maintain cyber vs. physical components independently. It is necessary instead to understand their interaction.

1.2. CPS structure

Figure 3 shows the structure of a Cyber-Physical System (CPS) schematically. Within a manufacturing system, an embedded system in the sense of a CPS is integrated within physical systems, e. g. the machines. The embedded system includes sensors to gather physical data and electronic hardware as well as software to save, and analyze data. The results of the data processing are the foundation for an interplay with other physical or digital systems by means of actuators.

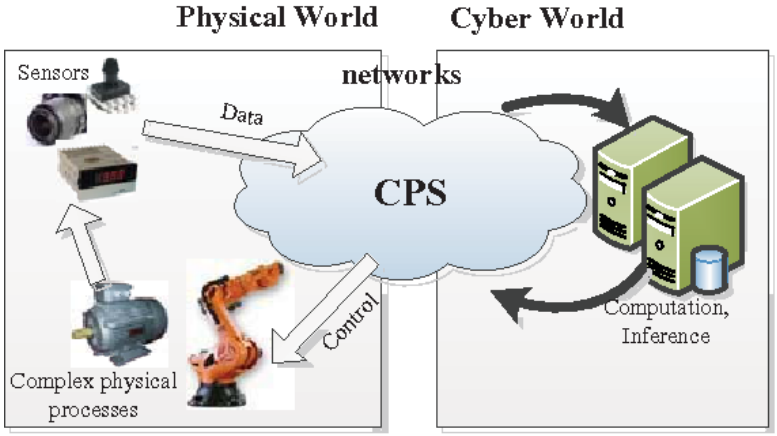


Fig. 3. A typical structure of the CPS

A CPS consists of one or more micro-controllers to control sensors and actuators which are necessary to collect data and interact from its environment. These systems also need communication interface to exchange data with other smart devices and a cloud. Data exchange is the most important feature of cyber physical systems. CPS connected over Internet are also known as Internet of Things.

CPS includes transdisciplinary approaches, combining the theory of cybernetics, mechatronics, design and process science. Process control is often called embedded systems.

Embedded systems are able to monitor and control physical processes by sensors and actuators. CPS are Embedded Systems, but are networked with each other to utilize globally or locally in another CPS available information sources and services. Accordingly, CPSs combine the vision of intelligent, adaptive control systems with seamless vertical, horizontal and dynamic information exchange between heterogeneous platforms (Gengarle et al., 2013).

Thus, CPS are a combination of interacting embedded computers and physical components. Both computation and physical processes work in parallel to bring about the desired output. Computers usually monitor the physical processes via sensors in real-time and provide feedback to actuators.

3. CYBER-PHYSICAL PRODUCTION SYSTEM

Production systems that already have computer technology are extended by network connection (Świć & Gola, 2013). They allow communication with other devices and output information about them. This is the next step in production automation. Networking of all systems leads to "Cyber-Physical Production Systems – CPPS", and thus to intelligent factories, in which production systems, components and people communicate through a network and production is almost autonomous. The system consisting of data, artificial intelligence, machines and communication is not only automated but also intelligent (Szabelski, Krawczuk & Domińczuk, 2014). The machine is able to collect data, analyze it and make decisions based on this analysis.

The definition of CPPS is from (Cardin, 2019): "Cyber-Physical Production Systems are systems of systems of autonomous and cooperative elements connecting with each other in situation dependent ways, on and across all levels of production, from processes through machines up to production and logistics networks, enhancing decision-making processes in real-time, response to unforeseen conditions and evolution along time".

The main differentiating requirements within a CPPS are: adaptability, convertibility, and integrality (Fig. 4). At the core of I4.0 lies the idea of constantly adapting systems. Adaptation can happen in structure, function, or both. As such, adaptation can only be implemented if the system components can be integrated with each other (integrality). It additionally requires a relative modular physical structure (convertibility) to support a wide scope of adaptive solutions beyond simple functional adaptation (Vogel-Heuser, Lee & Leitão, 2015).

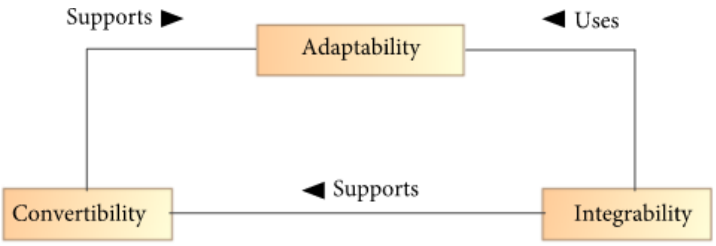


Fig. 4. Associations between the core requirements in CPPS

The extent to which different adaptability, convertibility, and integrality functions are implemented is a direct measurement of the degree of how cyber-physical, at the light of I4.0, a production system really is (Al-Alia, Guptab & Nabulsic, 2018).

The adaptability requirements focus mainly on the expectations on system behavior. It has been accepted for many years now that the ability to adapt to changing conditions is of paramount importance for production systems. The notion of CPPS seems to encompass also the possibility of structural adaptation whereby mobile equipment can even change, in a more or less autonomous way, the factory layout.

The convertibility requirements adaptation is reflected on how the system behaves. Convertibility is about the physical characteristics of the system that ultimately allow it to make use of its adaptive behavior. Modularity is greatly recognized as the prevailing characteristic, and a CPPS should ensure that its components can be combined in different ways to adapt and generate new functions when required. It therefore requires from its CPPM (Cyber-Physical Production Modules) a minimal level of mechanical interfacing and compatibility (Klimeš, 2014).

4. CONCEPTUAL MODEL OF CYBERNETIC PHYSICAL PRODUCTION SYSTEM

The conceptual model of the Cyber-Physical Production System consists of five layers: physical, network, data, analytical and application. The structure of the proposed model was presented in Fig. 5.

Physical layer: This layer consists of sensors, actuators, monitoring devices and computational elements. The real-time data collected from the product sensors can be processed locally by the operator and/or transferred to the cloud for further processing. Based on the system nested processing algorithm, the generated command to command the controls can be executed locally or remotely (Huebner, Facchi, Meyer & Janicke, 2013).

Network layer: CPS and CPPS can access cyberspace using various network protocols such as WiFi, WiMAX, GPRS and 3G/4G/LTE technology. Other IoT-oriented data protocols, such as MQTT, CoAP, AMQP, Websocket, and Node, are used to transfer data from peripheral devices to the cloud for further storage and processing. Each protocol has its advantages over others depending on speed, latency, bandwidth, reliability, security, and scalability.

Storage layer: CPPS systems collect a lot of data from objects that are in the physical layer. This data can be stored on a local server or in the cloud.

Processing and analytical layer: The processing and analytical layer is used to process data using simulation models (Gola & Świć, 2013). With the help of SQL queries, reports, graphs and visualizations, it is possible to generate data for monitoring purposes in almost real time. Data mining techniques such as data aggregation, classification, and regression can be used for predictive maintenance and planning. In this layer, monitoring and control actions can also be transferred back to the physical layer so that some devices and machines can be activated.

Application layer: This layer is the user interface for consumers, operators, manufacturers, third party suppliers and other service providers. It has a user-friendly access to an interface in which the above stakeholders can interact with the CPS layers based on privileged access and priority.

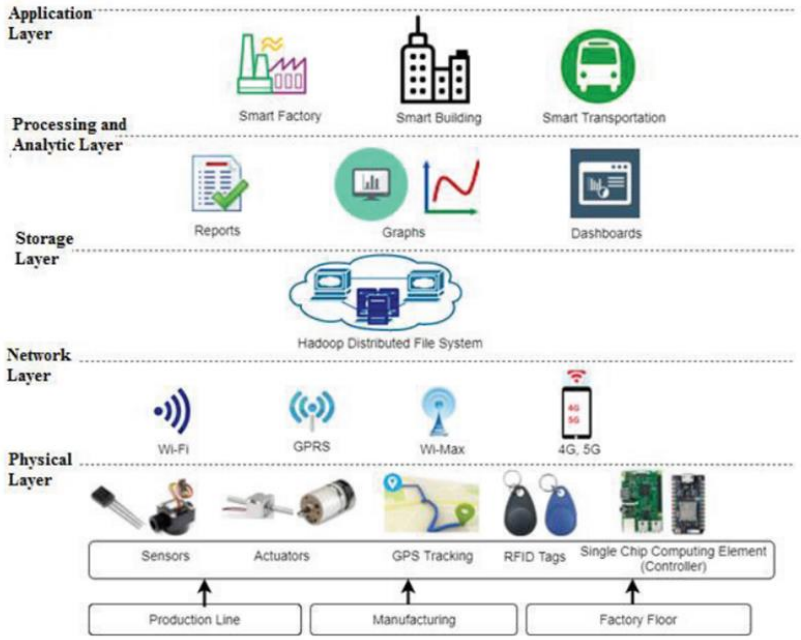


Fig. 5. Cyber Physical production System Conceptual Model

To verify the concept of the proposed model, it was implemented in a real production environment. A detailed description of the proposed solutions as well as the work carried out is presented in Chapter 5.

5. PRELIMINARY STUDIES – CASE STUDY

5.1. Characteristics of the enterprise

Based on the assumptions of the presented concept of a cyber-physical production system, solutions were implemented in a selected manufacturing enterprise dealing with the production of broadly understood fastening systems for the audio-video industry (Fig. 6). Manufacturing processes in a given enterprise are mainly carried out on CNC machines, and the majority of the processes involve machining.

The implementation of the system was aimed at acquiring significant parameters of key technological machines. The collected data allow to obtain a lot of important information that will be used in the process of the machines utilization optimizing and increasing the profitability of production (Gola, 2014).

Fig. 6. An example of the produced assortment

5.2. Implemented monitoring system

The implemented monitoring system was built using components of the solution called COMODIS, which is a wireless monitoring system (ASTOR, 2021). Additionally, the system uses selected industrial automation devices and a computer with appropriate software.

The proposed solution was built in keeping with the structure presented in Chapter 4. Due to the components used, the developed system combined the following layers:

- **Physical Layer** and **Network Layer** – mainly through the use of wireless analyzers communicating with the controller and made data available to the PLC controller (by means of appropriate network protocols),
- **Storage Layer** and **Processing and Analytical Layer –** through the use of a PLC controller that collected data from the controller and then made it available in the form of files easily interpreted by the computer software.

The diagram showing the connections of individual layers and the information flow in the system was presented in Fig. 7.

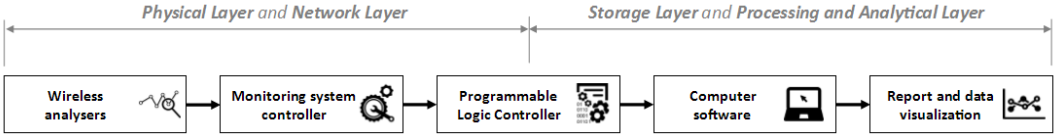


Fig. 7. Diagram of the implemented system

The idea of building the system was to use it in terms of obtaining information on the of the technological machines utilization that include:

- machine working times – determining the degree of machines utilization in total and on individual shifts,
- electricity consumption – constituting the basis for estimating the cost of machines utilization.

In the presented system 5 analyzers (measuring sensors) have been utilized. They have been installed in the control cabinets of the following technological machines:

1. Bending Center.
2. Punching Machine.
3. Press Brake Machine 1.
4. Press Brake Machine 2.
5. Laser.

Mentioned machines are crucial in the production – almost all production processes in the selected enterprise begin from these stations. The scheme of implementation and communication of the system components is presented in Fig. 8.

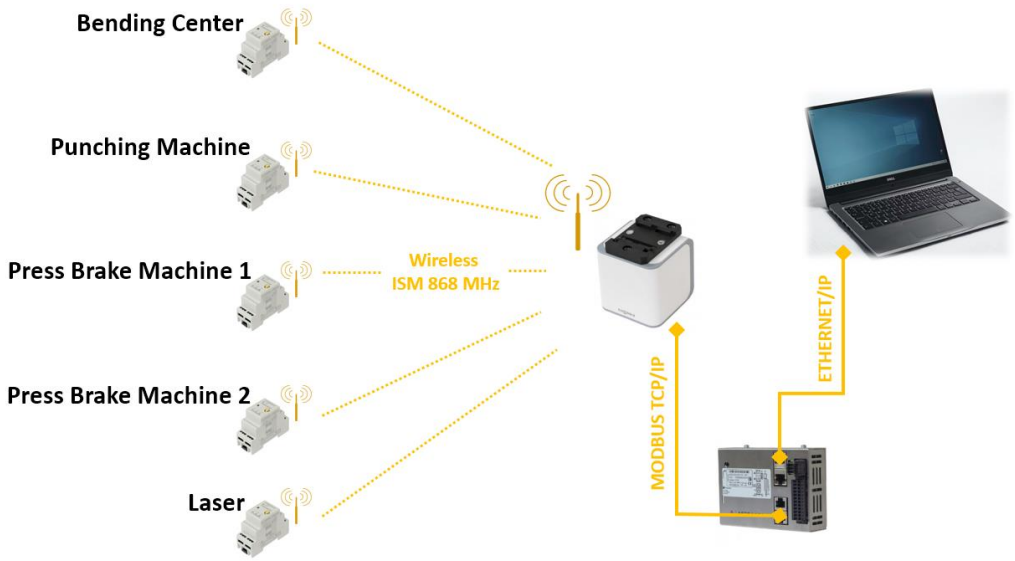


Fig. 8. The system and its components

The main components were the end elements of the system in the form of wireless energy analyzers, which (through the use of transformers) enable the monitoring of the parameters of a three-phase network with a neutral wire. The analyzers communicate with the controller by radio in the 868 MHz band, enabling the monitoring of 30 parameters. Selected parameters of the analyzers are presented in Table 1 (ASTOR, 2020b).

Tab. 1. Basic parameters of the analyzers (ASTOR, 2020b)

Energy analyzer	
Parameter	**Value**
Nominal supply voltage	230 V AC
Power supply frequency	50 Hz
Accuracy of measurement	0.5%
Transmission	Radio – ISM 868 MHz
Transmission method	Bidirectional – 9600 bps, 200 kbps
Working temperature range	0 to + 35 ° C
Mounting method	TH35 (DIN)
Transformer – primary current	100 A
Transformer – secondary current	33.3 mA
Antenna – cable length	3 m
Antenna – connector	SMA

In the presented system, the controller collects information about electricity parameters, but it is possible to expand it with additional sensors that allow you to control, for example, the temperature or the level of lighting. The selected parameters of the analyzers are presented in Table 2 (ASTOR, 2020a).

Tab. 2. Basic parameters of the controller (ASTOR, 2020a)

Controller	
Parameter	**Value**
Nominal supply voltage	5 V DC / 2 A Standard Micro-USB
Power consumption	1.1 W
Operating range	Up to 350 m outdoors
Connectors	RJ45 Ethernet Port, USB micro B 2.0, USB A 2.0
Maximum number of devices (end elements)	~ 255
Temperature range of operation	-10 to + 55 ° C
Mounting method	TH35 (DIN) or free standing
Communication	Radio – ISM 868 MHz (bidirectional – 9600 bps, 200 kbps), Modbus TCP
Analysis of additional parameters	temperature, light intensity

The advantage of the solution is a built-in web application that allows to manage end devices and observe the recorded values of the parameters (Fig. 9).

The controller provides communication via the Modbus TCP protocol with external devices. This possibility was used to integrate subsequent layers of the proposed system. In order to collect the registered data, the Astraada ECC2100 Slim PLC was used. The mentioned PLC is equipped with 4 digital inputs and 4 outputs, 4 analog inputs and communication modules: RS232/RS485 port, 2 configurable Ethernet cards. Additionally, it is equipped with a WebServer, USB port and Micro SD slot (ASTRAADA, 2015).

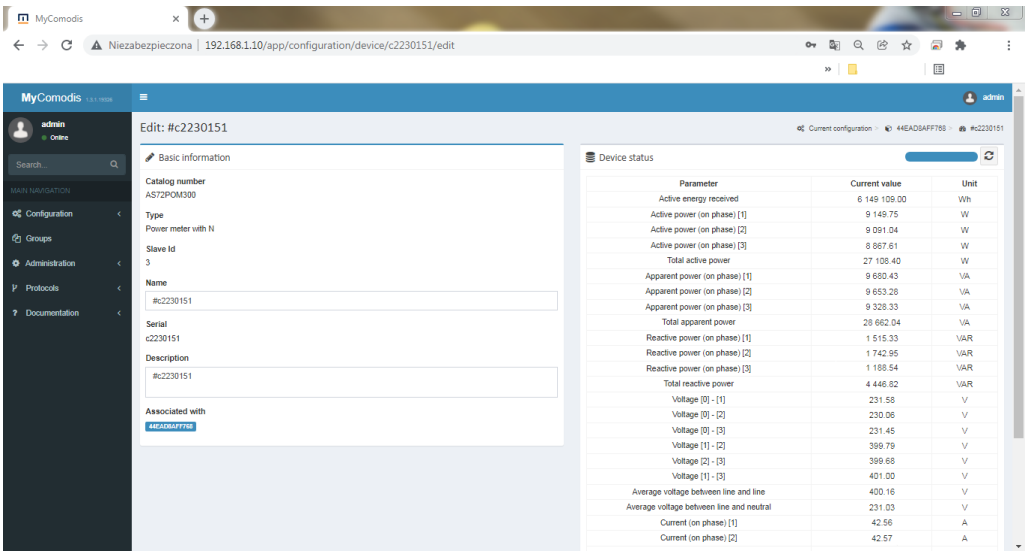


Fig. 9. Access to the main controller from the webservice level

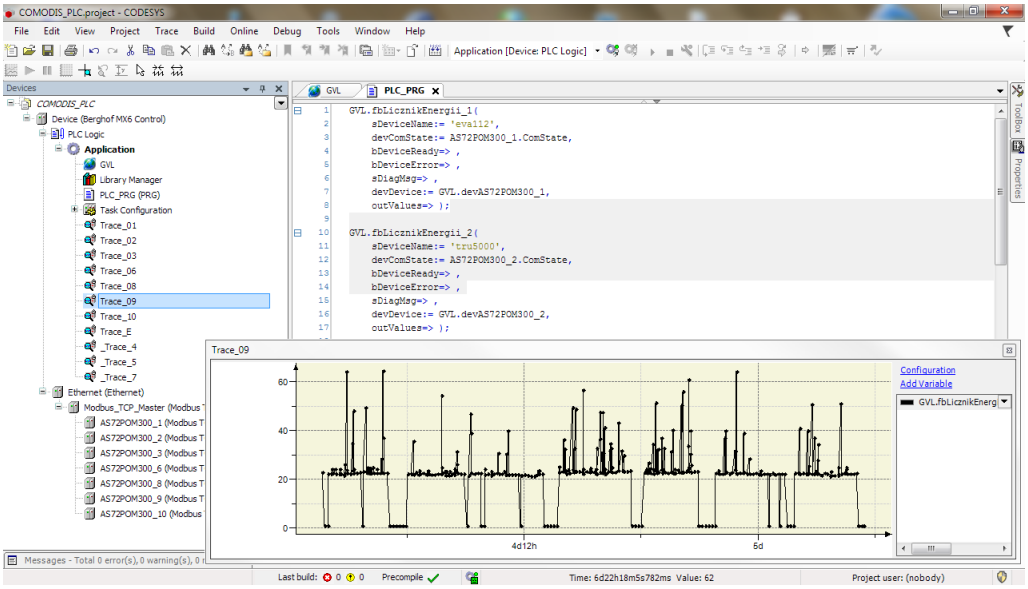


Fig. 10. Data collection using PLC and CODESYS environment

The CODESYS V3.5 (SP15 Patch 2) environment was used to program the controller, which was also used to implement the libraries enabling communication with PLC. As a result, it was possible to save the data recorded by the end elements of the system. Data was recorded at a frequency of 0.1 Hz. The data collection process by means of *Trace* component of the CODESYS environment is shown in Fig. 10.

The data collected with the PLC was exported to a *.CSV files. The use of this format enables convenient data exchange between various computer applications. In the proposed solution, the data was imported to a spreadsheet in which (through the use of appropriate

formatting and formulas) the data is presented to the user in a more accessible form (Fig. 11). It is important that through the use of the CSV format, the data can also be used in other specialized applications – for example, the Matlab or the RStudio environment.

RAW DATA	PROCESSED DATA		CHART			
Current [A]	Shift	Hour	Current [A]	Hour	Current [A]	Status
22.375	3	00:00:00	22,38	00:05:00	22,20	1
22.375	3	00:00:10	22,38	00:10:00	22,31	1
22.375	3	00:00:20	22,38	00:15:00	22,17	1
22.25	3	00:00:30	22,25	00:20:00	22,17	1
22.25	3	00:00:40	22,25	00:25:00	22,44	1
22.25	3	00:00:50	22,25	00:30:00	0,84	1
22.25	3	00:01:00	22,25	00:35:00	0,84	1
22.25	3	00:01:10	22,25	00:40:00	2,29	1
22.375	3	00:01:20	22,38	00:45:00	26,53	1
22.375	3	00:01:30	22,38	00:50:00	22,68	1
22.375	3	00:01:40	22,38	00:55:00	24,53	1
22.375	3	00:01:50	22,38	01:00:00	23,64	1
22.375	3	00:02:00	22,38	01:05:00	22,48	1
22	3	00:02:10	22,00	01:10:00	22,46	1
22	3	00:02:20	22,00	01:15:00	22,57	1
22	3	00:02:30	22,00	01:20:00	22,36	1
22	3	00:02:40	22,00	01:25:00	22,72	1
22	3	00:02:50	22,00	01:30:00	22,70	1
22.125	3	00:03:00	22,13	01:35:00	22,42	1
22.125	3	00:03:10	22,13	01:40:00	22,79	1
22.125	3	00:03:20	22,13	01:45:00	22,54	1
22.125	3	00:03:30	22,13	01:50:00	22,31	1
22.125	3	00:03:40	22,13	01:55:00	22,31	1
22.25	3	00:03:50	22,25	02:00:00	22,44	1
22.25	3	00:04:00	22,25	02:05:00	13,87	1
22.25	3	00:04:10	22,25	02:10:00	0,82	1
22.25	3	00:04:20	22,25	02:15:00	0,82	1
22.125	3	00:04:30	22,13	02:20:00	0,82	1
22.125	3	00:04:40	22,13	02:25:00	0,81	1
22.125	3	00:04:50	22,13	02:30:00	0,81	1
22.125	3	00:05:00	22,13	02:35:00	0,83	1

PARAMETERS	
Operating current [A]	10

MACHINE UTILIZATION:	
Idle time [h]:	22,5
Busy time [h]:	19,04
As a percentage [%]:	84,6%

UTILIZATION ON SHIFTS:	
SHIFT 1	
Idle time [h]:	7,5
Busy time [h]:	6,60
As a percentage [%]:	88,0%
SHIFT 2	
Idle time [h]:	7,5
Busy time [h]:	6,48
As a percentage [%]:	86,3%
SHIFT 3	
Idle time [h]:	7,5
Busy time [h]:	5,95
As a percentage [%]:	79,4%

PARAMETERS:	
Avg. voltage [V]	414,84
Avg. power factor [-]	0,79

ELECTRIC POWER CONSUMPTION:	
Daily [Wh]	140003,55
Daily [kWh]	140,00
Per hour [kWh]:	7,35

POWER CONSUMPTION:			
Hour	Current [A]	Power [W]	Inst. Cons. [Wh]
00:05:00	22,20	7275,46	606,29
00:10:00	22,31	7312,33	609,36
00:15:00	22,17	7264,54	605,38
00:20:00	22,17	7264,54	605,38
00:25:00	22,44	7353,30	612,77
00:30:00	0,84	0,00	0,00
00:35:00	0,84	0,00	0,00
00:40:00	2,29	0,00	0,00
00:45:00	26,53	8694,23	724,52
00:50:00	22,68	7433,86	619,49
00:55:00	24,53	8040,15	670,01
01:00:00	23,64	7747,93	645,66
01:05:00	22,48	7368,32	614,03
01:10:00	22,46	7360,13	613,34
01:15:00	22,57	7396,99	616,42
01:20:00	22,36	7327,35	610,61
01:25:00	22,72	7446,15	620,51
01:30:00	22,70	7439,33	619,94
01:35:00	22,42	7346,47	612,21
01:40:00	22,79	7469,37	622,45
01:45:00	22,54	7387,44	615,62
01:50:00	22,31	7312,33	609,36
01:55:00	22,31	7312,33	609,36
02:00:00	22,44	7353,30	612,77
02:05:00	13,87	4545,37	378,78
02:10:00	0,82	0,00	0,00
02:15:00	0,82	0,00	0,00
02:20:00	0,82	0,00	0,00
02:25:00	0,81	0,00	0,00
02:30:00	0,81	0,00	0,00
02:35:00	0,83	0,00	0,00

Fig. 11. Analysis of the collection of data with the use of a spreadsheet

The use of physical layer devices, the implementation of communication at the network level, as well as data collection and processing made it possible to verify the assumptions of the last layer of the system, i.e. the Analytical Layer. At this stage, reports were generated using the collected data. As a consequence, the conclusions were made and the areas of use of the obtained information were defined.

5.3. Use of data – results and conclusions

The data that was systematized and processed with the use of a spreadsheet was used to prepare reports concerning the analyzed machine operating parameters (time and energy consumption). A detailed report was prepared for each machine (Fig. 12), as well as a comprehensive data statement for all machines included in the system (Tab. 3). The results of analyzes prepared with the use of information obtained with the system for a period of one month are presented below.

Generating reports for each of the machines made it possible to determine the load level of each of them. Reports provide key information that is necessary for the effective implementation of production and optimization of the use of machines. For example – based on the report presented in Figure 12 – it can be concluded that with the use of Press Brake Machine 2 on shift 3 it was possible to implement additional work – the machine was occupied only for one half of the available time. This information can be used, for example, in the production planning department during the task scheduling process.

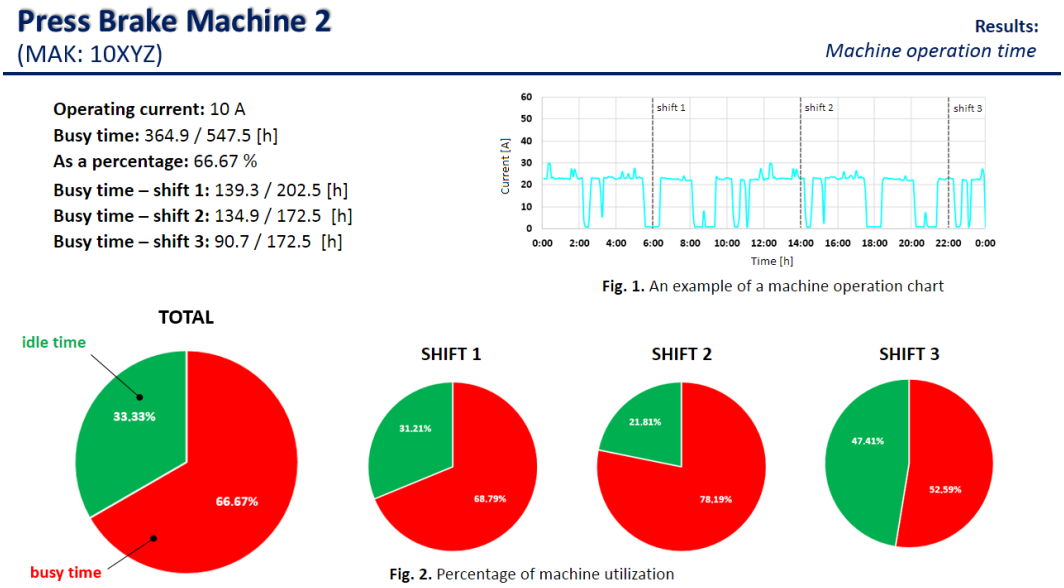


Fig. 12. Report generated based on collected data.

Tab. 3. Operation times for individual machines

ID	NAME	Busy time / available time [h]				Busy time / idle time [%]			
		Shift 1	Shift 2	Shift 3	TOTAL	Shift 1	Shift 2	Shift 3	TOTAL
01	Bending Center	29.0 202.5	73.0 172.5	23.6 172.5	**125.6** **547.5**	14.33 85.67	42.29 57.71	13.70 86.30	**22.97** **77.03**
02	Punching Machine	6.8 202.5	0.0 172.5	14.3 172.5	**21.1** **547.5**	3.35 99.65	0 100	8.31 91.69	**3.87** **96.13**
03	Press Brake Machine 1	129.9 202.5	112.8 172.5	135.2 172.5	**377.9** **547.5**	64.14 35.86	65.36 34.64	78.37 21.63	**69.21** **30.79**
06	Press Brake Machine 2	139.3 202.5	134.9 172.5	90.7 172.5	**364.9** **547.5**	68.79 31.21	78.19 21.81	52.59 47.41	**66.67** **33.33**
09	Laser	146.4 202.5	132.4 172.5	69.1 172.5	**348.0** **547.5**	72.28 27.72	76.76 23.24	40.14 59.86	**63.60** **36.4**

An overview of the operating times of all machines also provides a lot of information. These data can be used in the technology planning department. The variant technology allows some operations to be carried out with the use of other machines in order to relieve the machines with a high degree of load. An example is the use of the Punching Machine, which in the analyzed period was occupied only by 3.87% of the available time, while the Laser was used for 63.60% of the available time.

The implemented system also allowed for the analysis of electricity consumption by individual machines. The obtained results of the analyzes are presented in Table 4. The presented data are the basis for estimating the production costs on a given machine, and are also helpful in the process of orders valuation. Therefore, they provide a lot of information, valuable from the point of profitability of the production.

Tab. 4. Electricity consumption by individual machines

ID	NAME	ELECTRICITY CONSUMPTION [kWh] (for one month)	
		An average per hour	TOTAL
01	Bending Center	11.82	1486.34
02	Punching Machine	1.85	39.91
03	Press Brake Machine 1	1.13	428.55
06	Press Brake Machine 2	7.41	2705.68
09	Laser	10.72	3732.60

In order to fully implement the concept of a cyber-physical system, it is necessary to implement the system control in a closed circuit. Although it was not implemented in the area of the presented works, it is fully possible. Then, with the use of appropriate software and an expert system, it would be possible to automate processes and control production – for example by assigning orders based on the current load level, energy consumption or prediction based on obtained historical data. The conducted research proves that the models and concepts presented in this work are reasonable, and their use is the future of production systems.

6. CONCLUSION

It may still seem complicated, but cyber physical systems are complex. Therefore, if we want to understand Industry 4.0 or smart production, it is necessary to understand the essence of the basic technological pillars and the concept of new production, including CPS resp. CPPS, IoT, Big Data, artificial intelligence technologies and more. A significant CPS challenge involves defining and supporting new cooperative engineering paradigms to enable this synthesis of mechanical and software design and development. Physical systems are realized in matter, in contrast to logical systems conceptualized in software. In intersecting the two realms, cyber-physical systems are inherently harder to design, harder to model, harder to analyze, harder to simulate, harder to test, and therefore substantially more difficult to successfully innovate and realize. The implementation example of a monitoring system for selected parameters of technological machines operation presented in the paper confirms the advisability of the use of proposed solutions in practice. They provide a lot of information about the ongoing processes. The efficient data transfer and effective use of information is the basis of the technologies used in CPPS solutions. In the future, CPPS will be present in all industries and, under the Industry 4.0 paradigm, will open new production methodologies that will become tomorrow's industry standard.

Acknowledgements

The article is the result of projects:

- *Project number: 2019-1-SK01-KA202-060772. Title: TI4 –Technology Industry 4 for teachers and trainers of vocational education. Erasmus +. The article reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*
- *Project number: 010TUKE-4/2020. Project title: Implementation of new knowledge and innovative approaches to the process of teaching robotics in the intentions of Industry 4. KEGA.*

REFERENCES

ADDI-DATA. (2015, November 18). *CPS Cyber Physical Systems*. https://addi-data.com/cps-cyber-physical-systems

Al-Alia, R., Guptab, R., & Nabulsic, A. (2018). Cyber Physical Systems Role in Manufacturing Technologies. *AIP Conference Proceedings, 1957*, 050007. https://doi.org/10.1063/1.5034337

ASTOR. (2020a). *AS72CTR001: Instruction manual*.

ASTOR. (2020b). *AS72POM300: Instruction manual.*

ASTOR. (2021). *Bezprzewodowy, łatwy w integracji system monitoringu energii dla przemysłu*. COMODIS. https://www.comodis.pl

ASTRAADA. (2015). *ECC22XX Ethernet Controller Compact. User's Manual.*

Cardin, O. (2019). Classification of cyber-physical production systems applications: Proposition of an analysis framework. *Computers in Industry, 104*, 11–21. https://doi.org/10.1016/j.compind.2018.10.002

Gengarle, M. V., Bensalem, S., McDermid, J., Sangiovanni-Vincentelli, A., & Törngre, M. (2013). *Characteristics, Capabilities, Potential Applications of Cyber–Physical Systems: a Preliminary analysis*. CyPhERS Cyber-Physical European Roadmap & Strategy (Deliverable D2.1 – CPS Domain: Initial Synthesis).

Gola, A. (2014). Economic Aspects of Manufacturing Systems Design. *Actual Problems of Economics*, *156*(*6*) 205–212.

Gola, A., & Świć, A. (2013). Design of storage subsystem of flexible manufacturing system using the computer simulation method. *Actual Problems of Economics*, *142*(*4*), 312–318.

Harrison, R., Vera, D., Ahmad, B. (2016). Engineering Methods and Tools for Cyber–Physical Automation Systems. *Proceedings of the IEEE, 104*(5), 973–985. https://doi.org/10.1109/JPROC.2015.2510665

Huebner, A., Facchi, Ch., Meyer, M., & Janicke, H. (2013). RFID systems from a cyber-physical systems perspective. *Proceedings of the 11th International Workshop on Intelligent Solutions in Embedded Systems (WISES)* (pp. 1–6). IEEE.

i-SCOOP (2021). *Industry 4.0 and the fourth industrial revolution explained*. i-SCOOP. https://www.i-scoop.eu/industry-4-0

Klimeš, J. (2014). Using Formal Concept Analysis for Control in Cyber-physical Systems. *Procedia Engineering, 69*, 1518–1522. https://doi.org/10.1016/j.proeng.2014.03.149

Monostori, L. (2014). Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. *Procedia CIRP, 17*, 9–13. http://doi.org/10.1016/j.procir.2014.03.115

Onik, M. M. H., Kim, C., Yang, J. (2019). Personal Data Privacy Challenges of the Fourth Industrial Revolution. *21st International Conference on Advanced Communication Technology (ICACT)* (pp. 635–638). IEEE. http://doi.org/10.23919/ICACT.2019.8701932

Ratchev, S. (2017). *Cyber-Physical Production Systems*. Engineering and Physical Sciences Research Council. https://connectedeverythingmedia.files.wordpress.com/2018/06/cyber-physical-production-systems.pdf

Sabella, R. (2018, October 2). Cyber physical systems for Industry 4.0. Ericsson. https://www.ericsson.com/en/blog/2018/10/cyber-physical-systems-for-industry-4.0

Schuh, G., Potente, T., Varandani, R., Hausberg, C., & Fränken, B. (2014). Collaboration Moves Productivity to the Next Level. *Procedia CIRP, 17*, 3–8. http://doi.org10.1016/j.procir.2014.02.037

Strang, D., & Anderl, R. (2014). Assembly Process driven Component Data Model in Cyber-Physical Production Systems. *Proceedings of the World Congress on Engineering and Computer Science*. http://www.iaeng.org/publication/WCECS2014/WCECS2014_pp947-952.pdf

Świć, A., & Gola, A. (2013). Economic Analysis of Casing Parts Production in a Flexible Manufacturing System. *Actual Problems of Economics*, *141*(*3*), 526–533.

Szabelski, J., Krawczuk, A., & Dominczuk, J. (2014). Economic considerations of disassembly process automation. *Actual Problems of Economics*, *162*(*12*), 477–485.

Vogel-Heuser, B., Lee, J., & Leitão, P. (2015). Agents enabling cyber-physical production systems. *Automatisierungstechnik*, *63*(10), 777–789. https://doi.org/10.1515/auto-2014-1153

Yasniy, O., Pyndus, Y., Iasnii, V., & Lapusta, Y. (2017). Residual lifetime assessment of thermal power plant superheater header. *Engineering Failure Analysis*, *82*, 390–403. https://doi.org/10.1016/j.engfailanal.2017.07.028

Applied Computer Science, vol. 17, no. 4, pp. 100–109
doi: 10.23743/acs-2021-32

Submitted: 2021-07-12 | Revised: 2021-08-14 | Accepted: 2021-09-20

Keywords: computer clusters, parallel computing, n-body problem

Tomasz NOWICKI [0000-0003-0752-2509]*,
Adam GREGOSIEWICZ [0000-0002-6702-8505]**,
Zbigniew ŁAGODOWSKI [0000-0003-1811-6151]**

PRODUCTIVITY OF A LOW-BUDGET COMPUTER CLUSTER APPLIED TO OVERCOME THE N-BODY PROBLEM

Abstract

The classical n-body problem in physics addresses the prediction of individual motions of a group of celestial bodies under gravitational forces and has been studied since Isaac Newton formulated his laws. Nowadays the n-body problem has been recognized in many more fields of science and engineering. Each problem of mutual interaction between objects forming a dynamic group is called as the n-body problem. The cost of the direct algorithm for the problem is $O(n^2)$ and is not acceptable from the practical point of view. For this reason cheaper algorithms have been developed successfully reducing the cost to O(nln(n)) or even O(n). Because further improvement of the algorithms is unlikely to happen it is the hardware solutions which can still accelerate the calculations. The obvious answer here is a computer cluster that can preform the calculations in parallel. This paper focuses on the performance of a low-budget computer cluster created on ad hoc basis applied to n-body problem calculation. In order to maintain engineering valuable results a real technical issue was selected to study. It was Discrete Vortex Method that is used for simulating air flows. The presented research included writing original computer code, building a computer cluster, preforming simulations and comparing the results.

1. INTRODUCTION

The n-body problem arises occasionally in physics and thus also in engineering. The prerequisite for its emergence is (1) description of a physical phenomenon by means of a dynamic and discrete set of particles, which (2) influence mutually in the relationship "each with everyone". Computer modelling of physical phenomena in this way is simple and so attractive from an engineering point of view. However, the simplicity and purity of the method carries time–consuming calculations resulted from the necessity of recalculating all

* Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, Poland, t.nowicki@pollub.pl
** Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Mathematics, Poland, a.gregosiewicz@pollub.pl, z.lagodowski@pollub.pl

the mutual interactions at every step of the simulation. Engineers who applies n-body particle models in their practice always face the challenge of time-intensive computer calculation. It also should be noted that various particle models and methods (those in which the n-body problem occurs) are always accompanied by other additional complications that forms the individual computational specificity of them. Research presented in this article focuses on Discrete Vortex Method (DVM). The authors tried to answer the question what the efficiency of low-budget computer clusters can be when applied to DVM simulations.

1.1. The generalized n-body problem

Firstly, the n-body problem is going to be formulated in the simplest and general way (Fig. 1). For this purpose one should:

- define a metric space with a metric d and supply it with time t,
- spread at $t = 0$ a finite set of particles (called a discrete population or a discrete system) numbered $i = 1, 2, 3, \ldots, n$ in the space by determining their initial positions P_i and velocities u_i,
- abstract one common attribute of the particles, which intensity C determines the strength of mutual influence,
- chose a mutual influence function Q, which lets calculate influence from the particle j on i $Q_{ij} = Q(C_i, C_j, d_{ij})$,
- choose a velocity function u which lets calculate change in the velocity of the particle i $\Delta u_i = u(u_i, C_i, Q_1, Q_2, Q_3, \ldots, Q_n)$,
- choose a displacement function D, which lets calculate the change in a particle position after time Δt: $\Delta d_i = D(u_i, C_i, \Delta t)$,
- let the particles change their positions with time.

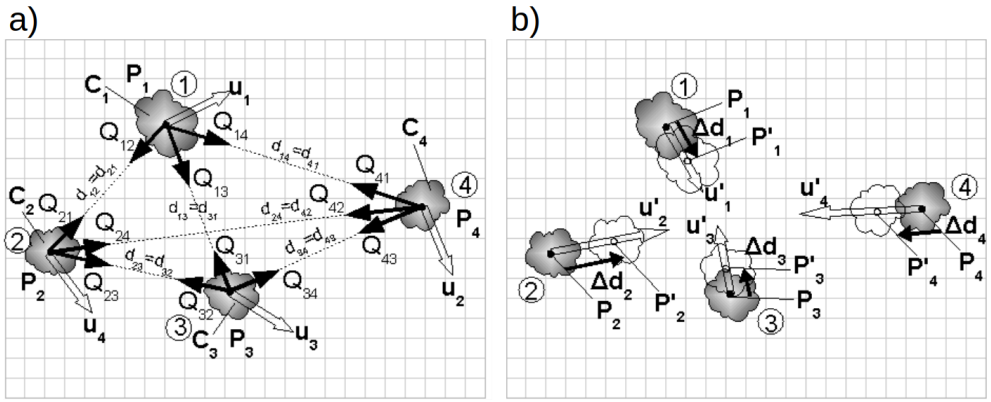


Fig. 1. A four-element discrete population (n = 4) in the "each to everyone" relation: a) the initial configuration, b) the change under mutual interactions; the description: i = 1, 2 ,3, 4 – particle numbers, C_i – the intensity of a attribute , P_i P_i' – the location in the space (before and after the change), u_i u_i' – velocity, Q_{ij} – the influence on particle i from j, Δd_i – the location change

Such a discrete population constantly reconfigures itself as time runs. All the particles continually influence each other and move in the space due to the influence. The movement results in a change of the mutual influences. The population equilibrium may or may not be achieved. Such a "numeric ecosystem" has the ability to reproduce a real phenomenon if constructed and interpreted in a correct way.

The n-body problem is considered resolved when is known the configuration of the population (positions and velocities of all the particles) at any time $t > 0$. It turns out that the n-body problem has in general no analytical solution. Subsequent configurations of the population may be determined only by direct simulations. There are three main groups of algorithms for n-body problem (Hockney & Eastwood, 1988): 1) particle-particle (P^2), 2) particle-mesh (PM) and 3) particle-particle-particle-mesh (P^3M). The P^2 algorithm is the simplest one and consists in calculating all single interactions in a direct way. It results in the numerical cost of $O(n^2)$, which is usually unacceptable in engineering practice. The PM algorithm employs a calculating mesh, which let reduce the cost to O(n) but also decreases accuracy of the results because small-scale local effects are not able to develop. The last P^3M algorithm is a combination of the previous two. For each particle a direct neighbourhood area is established within which the P^2 algorithm is used to calculate the influence from other particles from the neighbourhood. The influence from the remote particles are determined using the P^3M algorithm. The numerical cost of the last algorithm is $O(n \cdot ln(n))$. It is most frequently implemented and was used in the presented research.

1.2. The Discrete Vortex Method

The Discrete Vortex Method (DVM) (Lewis, 1991; Cottet & Koumoutsakos, 2000) is one of methods dedicated to computer simulating of turbulent fluid flows. The method was originated in the thirties of the twentieth century and has been applied successfully to fluid mechanics since then (Fig. 2).

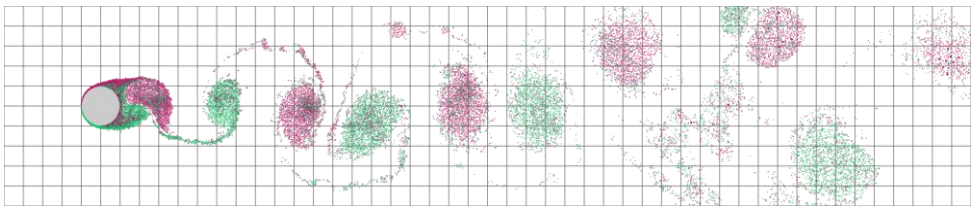


Fig. 2. Turbulent air flow over a cylinder with correctly developed vortex street as an example of DVM in action (Nowicki, 2012)

DVM is a numerical method developed for solving the Navier-Stokes equation (N-S) based on the Lagrangian model of a particle tracing. In DVM, the equation is solved by a direct computer simulation of a physical phenomena. A finite mesh known from finite element or finite volume methods is not applied in DVM. Artificial models of turbulence such as LES or k-ε are also not used. The most valuable feature of the method is its self-adaptability to geometry of computational task (Nowicki, 2015) and numeric stability. The biggest drawback of DVM is time consuming simulations come from the n-body problem.

Considering 2D euclidean areas of fluid flow and assuming a homogeneous dry air with a constant density, the following form of the N-S equation can be used to describe the phenomenon under interest:

$$\frac{\partial u}{\partial t} + (u\nabla)u = -\frac{1}{\rho}\nabla p + \nu\nabla^2 u \tag{1}$$

where: u – velocity field,
$(u\nabla)$– operator of the material derivative,
p – pressure field,
ρ – density of air,
ν – kinematic viscosity of air,
t – time.

Eq. (1) can be decomposed by calculating the rotation of the vector $\boldsymbol{u}$, which gives the so-called vorticity transport equation:

$$\frac{\partial \omega}{\partial t} + (u\nabla)\omega = \nu\nabla^2\omega \tag{2}$$

where: $\omega = \nabla \times u$ – vorticity field of the flow (treated as scalar for 2D flows).

The last eq. (2) is composed of two components: advection (3) and diffusion (4):

$$\frac{\partial \omega}{\partial t} + (u\nabla)\omega = 0 \tag{3}$$

$$\frac{\partial \omega}{\partial t} = \nu\nabla^2\omega \tag{4}$$

The separation (known as *Split Algorithm*) lets us treat the fluid flow as two simultaneous and independent phenomena: advection and diffusion, wherein only advection eq. (3) describes the vortex kinematics that leads to the n-body problem.

In DVM the computational particle is a discrete vortex. The abstracted attribute of the particle is its vorticity traditionally denoted by the letter Γ. The vorticity equals the value of circulation of velocity field over a contour (with element $\boldsymbol{dr}$) of an area from which the vorticity is reduced to a single point:

$$C = \Gamma = \oint_L u\, dr \tag{5}$$

The mutual influence function Q from particle j on i is given by a formula:

$$Q_{ij} = \Gamma_j \cdot K \times d_{ij} \tag{6}$$

where: $d_{ij} = P_j - P_i$- distance between vortexes as the metrics,
K – kernel articulating inverse-square law.

Since the influence function in DVM describes velocity field, there is no need to introduce an extra velocity function and:

$$u_i = \sum_{j=1,2,3,\ldots,n \wedge i \neq j} Q_{ij} \tag{6}$$

After the short glimpse of DVM given above it should be clear that the method incorporates the n-problem method.

1.3. Literature review

The results presented in this paper concern a numerical experiment carried out in 2007 (Nowicki, 2007). The aim of that experiment was to determine the performance of a low-cost computer cluster dedicated to DVM simulations. At the time, the method was in its early stage of development and such data was lacking. Today in 2021 the method can be still characterized as academic one because neither commercial nor open source software has been released yet. The interest of the method has not stopped as well, but its development is rather slow. In the period of 2007–2012 several hundred scientific papers on the subject have been published. About 300 can be found in the Scopus database, 100 in SpringerLink and 200 in ScienceDirect. The majority of published works concerns engineering applications of DVM or improving its accuracy. The problem of accelerating calculations appears extremely rarely and relates to modification of DVM algorithms rather than parallelization of calculations. And so, for example, Ricciardi, Wolf & Bimbato, 2017 studied the combination of exponential and power series expansions implemented using a divide and conquer strategy to accelerate the calculation while two years earlier he proposed fast multipole method algorithm to accelerate the expensive interactions of the discrete vortices (Ricciardi et al., 2015). The results of analysis on possibility of using fast matrix multiplication methods for the approximation of the velocity field when solving the system of differential equations describing the vorticity transport in an ideal incompressible fluid in Lagrangian coordinates can be found at Aparinov & Setukha (2009). Whereas Dynnikova (2009) explored the construction of a hierarchical structure of regions (tree) in order to accelerate the calculations. A different approach represents Huang, Su & Chen (2009), who introduced a concept of residual circulation in that sense that only a partial circulation of the vortex sheet is diffused into the flow field. The cited examples show that accelerating calculations with hardware methods has not been of interest to the researchers. Only Kuzima, Marchevsky & Moreva (2015) studied the speed-up in DVM calculations on multicore (using MPI and OpenMP) and graphic workstation (CUDA). She reported acceleration in calculations up to 40 times. On the other hand the interest in the classic n-body problem itself has not stopped. Despite the fact that today it is a well-recognized problem novel simulations are being preformed (e.g. Groen, Zwart, Ishiyama & Makino, 2011) and new software is being developed (e.g. Incardona, Leo, Zaluzhny, Ramaswamy & Sbalzarini, 2019).

Taking into account the above information any practical study on usage of computer clusters in the DVM should be in the field of interest of so called theoretical engineers. It happens very often that small research groups (at universities or in start-ups) ask themselves if it is worth to invest their time in building a computer cluster and creating parallel solvers in order to speed up calculations. The aim of this paper is to facilitate the answer to such questions in the case of Discrete Vortex Method. In this respect, the presented results remain still valid.

2. THE COMPUTER EXPERIMENT

The experiment was carried out in 2007 in a computer laboratory at Lublin University of Technology. The laboratory was equipped with 12 single-processor PCs connected with Fast Ethernet network. All the computers had the *AMD Atlon XP 1600+* 1.6GHz processor and 256MB RAM. The cluster was a symmetric one and was build according to Soan 2005. The *Ubuntu Linux 6.10* was used as an operating system and the *Mpich 2.0* as a communication layer. As a part of the experiment, three original DVM solvers were developed (see Suplement): *vorsym_s*, *vorsym_q* and *vorsym-p*. The program *vorsym_s* (vortex simulator slow) is a single-process and single-threaded program which implements the PP algorithm. The *vorsym_q* (quick) is also a single-process and single-threaded program but it implements the P^3M algorithm. Whereas the *vorsym_p* (parallel) solver is a multi-process (but still single-threaded) solver implementing the P^3M algorithm. The last program was run on the computer cluster using 4 or 9 nodes of it. (The number of nodes has been added in round brackets.) From the engineering point of view, the most important thing was to compare the execution time of calculations between *vorsym_q* along with *vorsym_p(4)* and *vorsym_p(9)*.

Tab. 1. The set of data used to perform the simulations

No.	File	Vortexes	Size		No.	File	Vortexes	Size
1	9.vrt	9	14 MiB		18	30k.vrt	29 929	45 GiB
2	25.vrt	25	38 MiB		19	40k.vrt	40 000	60 GiB
3	36.vrt	36	55 MiB		20	50k.vrt	49 729	74 GiB
4	49.vrt	49	75 MiB		21	60k.vrt	59 536	89 GiB
5	81.vrt	81	124 MiB		22	70k.vrt	69 696	104 GiB
6	100.vrt	100	153 MiB		23	80k.vrt	79 945	119 GiB
7	200.vrt	196	299 MiB		24	90k.vrt	90 000	134 GiB
8	300.vrt	289	441 MiB		25	100k.vrt	100 489	150 GiB
9	400.vrt	400	610 MiB		26	200k.vrt	200 704	299 GiB
10	500.vrt	484	739 MiB		27	300k.vrt	299 209	446 GiB
11	600.vrt	576	879 MiB		28	400k.vrt	399 424	595 GiB
12	700.vrt	676	1032 MiB		29	500k.vrt	499 849	745 GiB
13	800.vrt	784	1196 MiB		30	600k.vrt	600 625	895 GiB
14	900.vrt	900	1373 MiB		31	700k.vrt	700 569	1044 GiB
15	1k.vrt	1024	1563 MiB		32	800k.vrt	801 025	1194 GiB
16	10k.vrt	10 000	15 GiB		33	900k.vrt	900 601	1345 GiB
17	20k.vrt	19 881	30 GiB		34	1M.vrt	1 000 000	1.5 TiB

For the experiment 34 numerical samples were generated. They were files defining the initial conditions of the n-body DVM tasks. The samples differed in the number of vortex particles (Tab. 1). In each case the size of the computational space (domain) was of the same size of 100×100. Initially the vortexes were randomly and evenly distributed in the domain. The random Marsagil generator was used. The vortexes had also random strengths from –1.0 to 1.0. All the initial velocities were zero. All simulations were carried out with a constant step time equals to 0.01. The number of vortices in the domain was fixed. Vortices that crossed the domain boundary making their moves were returned to the domain from the opposite edge in such way that they continued their movement on the opposite side. The column *Size* gives the sizes of the output files for a 100 000 step simulation for each case.

3. RESULTS

The main aim of the simulations was to test the efficiency of the developed computational system considered as the computer cluster and dedicated solver *vorsym_p*. The simulations were carried out for all prepared files (Tab. 1). Depending on the size of a task a single simulation took form 3 to 10 000 steps due to time constraints. In order to normalize the obtained results, the average execution time of a single step was calculated. Results has been presented in a table (Tab. 2) and in a diagram (Fig. 3). The specimen numbers from Tab. 1. agrees with numbers from Tab. 2. Simulations for the specimens number form 22 to 34 were not performed with *vorsym_s* due to too long calculating times. Additionally, for the for *vorsym_q* and *vorsym_p* simulators number of computing subdomains were given. The subdomains were formed by dividing the square main domain to, also square, areas. The sides of the main domain were divided as follow: 2×2, 3×3, 4×4, …, 19×19, which resulted in 4, 9, 16, …, 361 subdomains respectively. They determined the calculating mesh of the P^3M algorithm. For the parallel *vorsym_p* solver firstly the main domain was divided into subdomains of distinct processes (4 or 9) then each process created its own subdomains according to the previously described rule.

A typical engineering problem solved using DVM requires at least tens of thousands discrete vortexes and performing about 100 000 simulation steps. Approximate times of completing such tasks were estimated on the basis of results from Tab. 2 and presented in Tab. 3. The results were valid in 2007 (for hardware reasons) but still clearly show significant reduction of computational time when a computer cluster is used for DVM. Whereas relative speed-ups of calculations presented on a diagram in the Fig. 4 has not outdated at all. It was noticed that 4 node cluster speeded up simulation 8 times and 9 node cluster – 22 times! The results are dubious but absolutely correct. So why did cooperation of n nodes caused acceleration greater than n times? The answer is the large amount of data generated at each calculating step and written into files. In the case of n nodes there were n different files on different hard dives instead of one big file on a single drive, which shortened the execution time of each step. Each node wrote only its own data. It was the amount of output data along with the specificity of n–body problem what determined the time of simulations. It is also the reason why obtained results does not correspond with those found in Kuzmina et al. (2015) where for small number of calculating cores linear acceleration was observed. Simply, the simulation time did not include data recording to files at each step of the simulation. This can not be avoided in engineering practice, which makes low-budget cluster very effective while deployed in DVM calculations. Such observation is very important to an engineer who has a task to shorten the time of DVM simulations as much as possible.

Tab. 2. Averaged time of performing a single simulation step for each file

No.	vorsym_s	vorsym_q		vorsym_p (4)		vorsym_p (9)	
	T [s]	N_{sub}	T [s]	N_{sub}	T [s]	N_{sub}	T [s]
1	0.00015	1	0.00029	1	0.00660	1	0.04860
2	0.00042	4	0.00072	1	0.00660	1	0.08000
3	0.00066	4	0.00091	1	0.00650	1	0.05000
4	0.00108	4	0.00111	4	0.00800	1	0.04900
5	0.00190	9	0.00146	4	0.00804	1	0.04800
6	0.00276	9	0.00158	4	0.00800	1	0.04600
7	0.01060	9	0.00280	4	0.01005	4	0.06670
8	0.02260	9	0.00490	4	0.01001	4	0.08230
9	0.04340	16	0.00803	9	0.01030	4	0.10076
10	0.06400	16	0.01100	9	0.01502	4	0.08330
11	0.09100	16	0.01403	9	0.01510	4	0.06000
12	0.14500	16	0.01900	9	0.01511	4	0.10040
13	0.16700	16	0.02201	9	0.01512	9	0.10108
14	0.21800	25	0.02700	9	0.01540	9	0.10204
15	0.30082	25	0.03012	9	0.01750	9	0.10160
16	27.1	64	0.9	36	0.3	25	0.1
17	106.3	100	2.7	49	0.8	36	0.5
18	241.7	121	5.1	64	1.0	36	0.7
19	432.0	144	7.8	64	1.3	49	1.0
20	685.0	144	10.7	81	1.7	49	1.3
21	952.0	169	14.3	81	2.0	64	1.3
22	-	169	18.1	100	2.7	64	1.7
23	-	196	21.7	100	3.0	64	2.0
24	-	225	27.0	100	3.5	64	2.3
25	-	225	30.7	121	4.3	81	2.7
26	-	324	89	169	12	100	5
27	-	400	159	196	23	121	9
28	-	441	245	225	32	144	13
29	-	484	341	256	44	169	17
30	-	529	452	289	57	196	22
31	-	576	547	289	73	196	27
32	-	625	697	324	87	225	33
33	-	625	829	324	104	225	38
34	-	279	994	361	122	225	44

Tab. 3. Estimated time of completing a 100 000 step DVM simulation

Solver	Number of vortexes in the simulation		
	20 000	**50 000**	**100 000**
vorsym_s	4 months	2 years	9 years
vorsym_q	3 days	12 days	36 days
vorsym_p(4)	22 hours	2 days	5 days
vorsym_p(9)	14 hours	1½ day	3 days

Obtained results may seem outdated nowadays due to the development in computer hardware since 2007. It is evident that in 2021 the simulations, if recreated would be completed in much shorter times even using the same computer code. It would be simply achieved by using faster CPUs and hard disc drives. That said, it is also evident that the new hardware nowadays could be used for bigger problems. In other words the discussion today would consider bigger task. Since the nature of the n-body problem has not changed it is still the writing of the output to hard discs which delays calculations significantly. What could improve the performance in this area is using multicore CPUs, which let delegate the writing tasks to a separate threads. It should be undoubtedly the first idea to be explored. Another way to overcome the problem of time-consuming n-body simulations could be using the GPUs technology that is much more affordable now, though this question is beyond the scope of this paper.

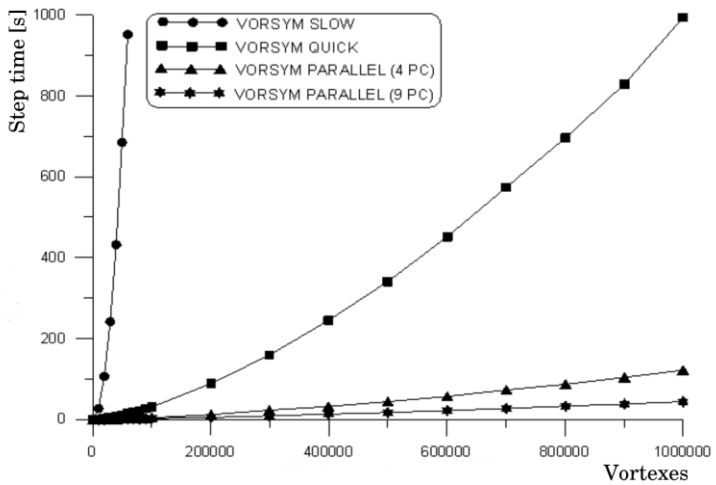


Fig. 3. Averaged time of performing a calculation step against the size of a task for different calculation methods

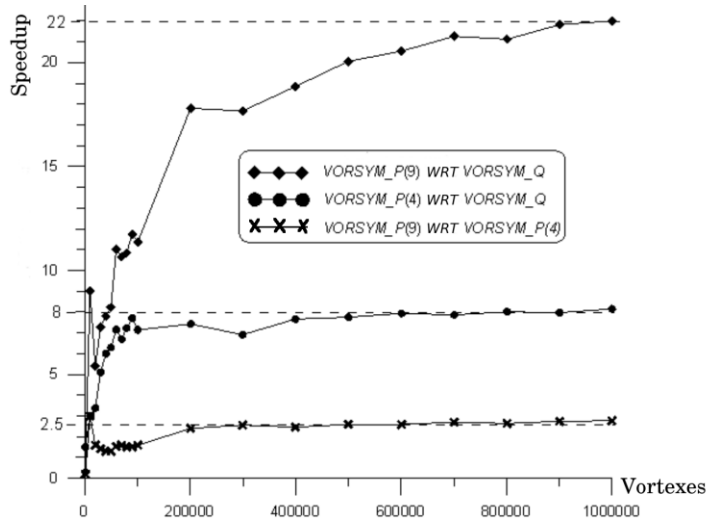


Fig. 4. Relative speedup of simulation (WRT abbr "with relation to")

4. CONCLUSIONS

In this paper results on the possibility of accelerating the Discrete Vortex Method computer simulations were presented. A low-budget computer cluster was build and a parallel solver was developed. Obtained acceleration of calculations exceeded the number of the cluster nodes due to division of the computing domain and separation of the output files. The paper deals with issues rarely described in the literature on the discrete vortex method.

Supplement

https://github.com/TomekNowicki/vorsym

REFERENCES

Aparinov, A. A., & Setukha, A. V. (2009). On the application of mosaic-skeleton approximations of matrices for the acceleration of computations in the vortex method for the three-dimensional Euler equations. *Differential Equations*, *45*, 1358. http://doi.org/10.1134/S0012266109090110

Cottet, G. H., & Koumoutsakos, P. D. (2000). *Vortex Methods Theory and Practice.* Cambridge University Press.

Dynnikova, G. Ya. (2009). Fast technique for solving the *N*-body problem in flow simulation by vortex methods. *Computational Mathematics and Mathematical Physics*, *49*, 1389–1396. http://doi.org/10.1134/S0965542509080090

Groen, D., Zwart, S. P., Ishiyama, T., & Makino, J. (2011). High Performance Gravitational N-body Simulations on a Planet-wide Distributed Supercomputer. *Computational Science & Discovery*, *4*(1), 015001. http://doi.org/10.1088/1749-4699/4/1/015001

Hockney, R. W., & Eastwood, J. W. (1988). *Computer Simulation Using Particles*. Taylor & Francis Group.

Huang, M. J., Su, H. X., & Chen, L. Ch. (2009). A fast resurrected core-spreading vortex method with no-slip boundary conditions. *Journal of Computational Physics*, *228*(6), 1916–1931. https://doi.org/10.1016/j.jcp.2008.11.026

Incardona, P., Leo, A., Zaluzhny, Y., Ramaswamy, R., & Sbalzarini, I. F. (2019). OpenFPM: A scalable open framework for particle and particle-mesh codes on parallel computers. *Computer Physics Communications*, *241*, 155–177. https://doi.org/10.1016/j.cpc.2019.03.007

Kuzmina, K., Marchevsky, I., & Moreva, V. (2015). Parallel Implementation of Vortex Element Method on CPUs and GPUs. *Procedia Computer Science*, *66*, 73–82. https://doi.org/10.1016/j.procs.2015.11.010

Lewis, R. I. (1991). *Vortex Element Methods for Fluid Dynamics of Engineering Systems.* Cambridge University Press.

Nowicki, T. (2007). *Algorytm równoległy dla problemu n-ciał* (Unpublished master thesis). Lublin University of Technology, Lublin. https://github.com/TomekNowicki/vorsym/blob/main/nowicki_n-body.pdf

Nowicki, T. (2012). *Wpływ sposobu realizacji warunków brzegowych w metodzie wirów dyskretnych na odpowiedź aeroelastyczną pomostów*. Politechnika Lubelska.

Nowicki, T. (2015). The Discrete Vortex Method for estimating how surface roughness affects aerodynamic drag acting on a long cylinder exposed to wind. *Technical Transactions, Civil Engineering*, *2-B*(12), 127–144. https://doi.org/10.4467/2353737XCT.15.129.4166

Ricciardi, T. R., Wolf, W. R., & Bimbato, A. M. (2017). A fast algorithm for simulation of periodic flows using discrete vortex particles. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *39*, 4555–4570. http://doi.org/10.1007/s40430-017-0902-x

Ricciardi, T., R., Bimbato, A. M., Wolf, W., R., Idelsohn, S. R., Sonzogni, V., Coutinho, A., Cruchaga, M., Lew, A., & Cerrolaza, M. (2015). Numerical simulation of vortex interactions using a fast multipole discrete particle method. *Proceedings Of The 1st Pan-american Congress On Computational Mechanics And Xi Argentine Congress On Computational Mechanics* (pp. 1065–1076). Barcelona: Int Center Numerical Methods Engineering.